

Frédéric GARDI

Maîtrise d'Informatique 1999-2000

Aix-Marseille II - Luminy

## TER sur le QUICKSORT

Etude de la complexité de quelques variantes de l'algorithme de tri Quicksort

Professeur responsable: Michel VAN CANEGHEM



## REMERCIEMENTS

Nous voudrions, avant toute chose, adresser nos remerciements à Mr Michel Van Caneghem, professeur d'Informatique à la Faculté des Sciences de Luminy, qui nous a encadré durant tout ce travail de recherche.



**SOMMAIRE**

INTRODUCTION ..... p.2

CHAPITRE I  
    La variante du Quicksort à t paliers: ses performances en moyenne ..... p.4

CHAPITRE II  
    Le Quicksort dans le pire des cas ..... p.11

CONCLUSION ..... p.15

ANNEXES ..... p.16

BIBLIOGRAPHIE ..... p.54



## INTRODUCTION

La méthode de tri Quicksort a été mise au point par C.A.R. Hoare en 1961. Elle a connu plusieurs évolutions au niveau algorithmique durant les années 60 et 70, qui lui ont permises d'améliorer son temps d'exécution déjà très rapide dans sa version initiale.

Voici un exemple d'implémentation en C d'une version basique du Quicksort que nous appellerons version (O):

```
void QS(int a[],int g,int d)
{
    int i,j,v,t;
    if (d>=g)
    {
        v=a[d];i=g-1;j=d;
        for (;;)
        {
            while (a[++i]<v);
            while (j>g && a[--j]>v);
            if (i>=j)
                break;
            t=a[i];a[i]=a[j];a[j]=t;
        }
        t=a[i];a[i]=a[d];a[d]=t;
        QS(g,i-1);
        QS(i+1,d);
    }
}
```

Le Quicksort est un algorithme de tri particulier. En effet, il s'exécute en  $O(n \log_2 n)$  dans le meilleur des cas, ce qui est très bon, mais en  $O(n^2)$  dans le pire des cas, ce qui est beaucoup moins bon.

Mais ce qui fait du Quicksort un des meilleurs algorithmes de tri est que son temps d'exécution en moyenne est en  $O(n \ln n)$  avec une constante cachée par la notation  $O$  faible. De plus, il n'utilise pas de tableaux annexes pour trier (il est "in-place"), il nécessite donc très peu de mémoire afin de s'exécuter. Il a tout de même un certain inconvénient: il n'est pas stable.

Comme nous le disions précédemment, l'algorithme du Quicksort a subi quelques évolutions lui permettant d'accroître ces performances. La version la plus connue et sûrement la plus utilisée jusqu'à aujourd'hui est le Quicksort avec un choix du pivot comme la médiane de trois éléments et une coupure à 10 éléments par le tri par insertion. Cette méthode a été étudiée en détail par R. Sedgewick (voir Bibliographie page 54). C'est d'ailleurs cette version qui a été implémentée sous Unix avec la fonction "qsort".

Mais cette variante est-elle vraiment la meilleure ? Peut-on en trouver une encore plus rapide voire même une qui soit optimale ? Le Quicksort peut-il s'exécuter en  $O(n \log_2 n)$  dans le pire des cas tout en restant compétitif avec d'autres algorithmes ?... Ce travail de recherche essayera de répondre à ces diverses questions.

Nous exposerons donc, dans le premier chapitre, des variantes du Quicksort (certaines ont déjà été étudiées, d'autres non) et nous analyserons leurs performances dans le cas moyen, en prenant comme hypothèse la distinction des clés à trier. Ces différentes analyses nous mèneront naturellement vers une variante encore plus performante du Quicksort.

Ces variantes sont le Quicksort médiane de  $2t + 1$  éléments sans coupure, le Quicksort médiane de  $2t + 1$  éléments avec coupure (par le tri par insertion) et enfin un Quicksort médiane de  $2t + 1$  éléments avec  $t$  coupures: le Quicksort "à  $t$  paliers". Nous appuyerons nos résultats généraux sur des exemples précis (Quicksort médiane de trois, médiane de cinq).

Pour faire une étude minutieuse de ces algorithmes, il nous faudra donc déterminer avec précision les paramètres dont vont dépendre leurs performances une fois qu'ils seront codés sur machine.

Ces paramètres, fonctions de  $n$  (le nombre d'éléments à trier), sont le nombre de comparaisons  $C_n$ , d'échanges  $E_n$ , de partitions  $P_n$ , d'appels  $A_n$  effectués par le Quicksort et la hauteur de pile  $Hp_n$  nécessaire au fonctionnement de son programme.

Le temps nous étant compté, nous n'avons calculé la hauteur de pile  $Hp_n$  que pour la version (O) du Quicksort. Nous obtenons  $Hp_n \leq 2 \log_2 n$ . Nous donnons le détail des calculs dans l'Annexe I pages 18 et 19. Pour la même raison, nous avons mis l'accent sur le calcul et la minimisation de  $C_n$  dans les cas généraux et donner les résultats correspondants pour  $A_n$  et  $P_n$  (plus simples). Nous n'étudierons pas en détail  $E_n$ ,  $E_n$  étant une combinaison linéaire de  $C_n$  et de  $A_n$ .

Par exemple, pour l'algorithme du Quicksort décrit ci-dessus, nous avons les résultats suivants: [R.Segdewick, P.Flajolet, "Introduction à l'analyse des algorithmes", 1996]

$$\begin{cases} C_n = 2(n+1)(H_{n+1} - 1) \approx 2n \ln n - 0.846n \\ E_n = \frac{1}{3}(n+1)(H_{n+1} - 5/2) + 1/2 \approx \frac{1}{3}n \ln n - 1.346n \\ A_n = 2n \\ P_n = n \end{cases}$$

Nous confirmerons bien entendu nos résultats théoriques par quelques expérimentations pratiques. Nous établirons aussi les points de coupure optimaux, pour chacun des algorithmes que nous citons en exemple, à partir de leurs temps d'exécution sur machine. Enfin, nous concluons sur l'intérêt pratique des méthodes exposées.

Dans le second chapitre, nous discuterons du Quicksort en  $O(n \log_2 n)$  dans le pire des cas: comment y-parvient-on? Les résultats obtenus sont-ils satisfaisants? Et quelles conclusions pouvons-nous tirer de cette étude? Nous utiliserons pour cela quelques résultats intéressants sur les algorithmes de sélection.

Pour introduire cette variante, nous allons tout d'abord analyser la variante du Quicksort avec médiane de  $2t + 1$  éléments dans une première partie, puis étudier les points de coupure par le tri par insertion dans une deuxième partie et enfin nous étudierons la version du Quicksort que nous avons appelé "à  $t$  paliers", qui est une généralisation de la méthode de la "coupure". Nous donnerons de plus pour chacune de ces parties des résultats plus précis sur les variantes médiane de trois et médiane de cinq dans l'Annexe.

### 1/ Les limites de la méthode de la "médiane" :

Nous avons analysé le Quicksort médiane de trois et médiane de cinq, voici les relations de récurrence relatives à chacun:

$$\text{Médiane de 3} \left\{ \begin{array}{l} C_n = n + 1 + \frac{8}{3} + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (C_{k-1} + C_{n-k}) \\ E_n = \frac{1}{5} \frac{n^2 - 2n + 2}{n-1} + 2 + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (E_{k-1} + E_{n-k}) \\ A_n = 1 + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (A_{k-1} + A_{n-k}) \\ P_n = 1 + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (P_{k-1} + P_{n-k}) \end{array} \right.$$

$$\text{Médiane de 5} \left\{ \begin{array}{l} C_n = n + 1 + 6 + \sum_{k=1}^n \frac{C_{k-1}^2 C_{n-k}^2}{C_n^5} (C_{k-1} + C_{n-k}) \\ E_n = \frac{1}{14} \frac{3n^2 - 5n + 6}{n-1} + 2 + \sum_{k=1}^n \frac{C_{k-1}^2 C_{n-k}^2}{C_n^5} (E_{k-1} + E_{n-k}) \\ A_n = 1 + \sum_{k=1}^n \frac{(n-k)(n-k-1)(k-1)(k-2)}{C_n^5} (A_{k-1} + A_{n-k}) \\ P_n = 1 + \sum_{k=1}^n \frac{(n-k)(n-k-1)(k-1)(k-2)}{C_n^5} (P_{k-1} + P_{n-k}) \end{array} \right.$$

La résolution de ces équations se trouve dans les Annexes II à IX (pages 20 à 29). La vérification pratique de ces résultats se trouve dans l'Annexe XVIII page 45 et 46.

Nous pouvons constater que ces deux versions sont plus efficaces que la version (O) du Quicksort. Mais que montrent réellement ces résultats ? Tout d'abord, le nombre d'échanges croît au fur et à mesure que l'on recentre le pivot. En effet, cela pourrait se démontrer en résolvant la récurrence du nombre d'échanges dans le cas général pour un Quicksort médiane de  $2t + 1$  éléments, qui est:

pour  $1 \leq t \leq \frac{n-1}{2}$ :

$$E_n = 2 + \sum_{k=1}^n \left( \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} \right) \frac{(n-k)(k-1)}{n-1} + \sum_{k=1}^n \left( \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} \right) (E_{k-1} + E_{n-k})$$

et pour  $t = 0$ :

$$E_n = 1 + \frac{1}{n} \sum_{k=1}^n (E_{k-1} + E_{n-k})$$

Nous avons fait une approximation du résultat de ce calcul en ne donnant que le coefficient du terme dominant. Mais cela nous suffit pour bien comprendre l'évolution du nombre d'échange en fonction de  $t$ . Nous détaillons cela dans l'Annexe X page 32 et le Graphe II de l'Annexe XIX page 48. L'autre fait important ici est que la recherche du médian accroît considérablement le nombre total de comparaisons, à tel point que le Quicksort médiane de trois reste meilleur que le Quicksort médiane de cinq, du point de vue du nombre de comparaisons effectuées. Ainsi la recherche du médian annulerait-elle les effets positifs du recentrage du pivot ? Par exemple, notre algorithme de la recherche du médian de cinq éléments effectue 6 comparaisons dans tous les cas. Mais ce n'est pas l'optimum en moyenne. En effet, D. Hoey, à qui nous avons emprunté cette procédure, a prouvé que l'on pourrait le faire théoriquement en  $5\frac{13}{15}$  comparaisons dans le cas moyen. Nous voyons ici que notre algorithme donne un résultat tout de même proche de l'optimum, et pourtant le nombre de comparaisons effectuées par celui-ci nuit aux performances de la version du Quicksort qui l'utilise. Cela pose le problème de l'optimalité de la méthode de la "médiane". Pour le résoudre, nous allons donc comme pour le nombre d'échanges, faire une généralisation à des échantillons de  $2t + 1$  éléments.

Il a été établi que le médian de  $n$  éléments peut être trouvé avec, en moyenne,  $\overline{M}_n = \frac{3}{2}n + O(n^{\frac{2}{3}} \ln n)$  comparaisons [R.W. Floyd, 1970].

Nous pourrions faire l'approximation  $\overline{M}_n = \frac{3}{2}n$  pour simplifier nos calculs. En effet, nous allons rapidement nous intéresser qu'à de petits échantillons pour la recherche de la médiane: nous pourrions alors utiliser un algorithme optimal de sélection du médian et le nombre de comparaisons effectué par celui-ci sera alors strictement inférieur à  $\frac{3}{2}n$  [D.Hoey (cf Knuth, "Sorting and Searching", 1973)].

La récurrence serait alors:

$$C_n = \frac{3}{2}(2t + 1) + n + 1 + \sum_{k=1}^n \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} (C_{k-1} + C_{n-k}) \quad (a)$$

$$\text{avec } 0 \leq t \leq \frac{n-1}{2} \quad \text{et} \quad n \geq 2t + 1$$

Pour l'analyser, nous devons considérer 2 cas que nous résoudrons séparément:

- pour  $t$  constant
- pour  $t$  dépendant de  $n$

1<sup>er</sup> cas :

Considérons  $t$  constant et posons  $\lambda = \frac{3}{2}(2t + 1)$  et résolvons les équations:

$$C_n = n + 1 + \sum_{k=1}^n \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} (C_{k-1} + C_{n-k}) \quad (b)$$

et

$$C_n = \lambda + \sum_{k=1}^n \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} (C_{k-1} + C_{n-k}) \quad (c)$$

L'équation (b) a été résolue par Van Emden (1970) [cf Knuth, "Sorting and Searching", 1973] mais nous l'avons repris afin de préciser le résultat. Puis, nous avons résolu (c) suivant le même schéma (voir Annexe X pages 31 et 32).

En combinant linéairement les solutions de (b) et de (c), nous obtenons la solution de (a), soit:

$$C_n = \alpha(n + 1)(H_{n+1} - 1) + \lambda n + K_0(n + 1) \quad \text{avec } \alpha = \frac{1}{H_{2t+2} - H_{t+1}} \quad \text{et} \quad \lambda = \frac{3}{2}(2t + 1) \quad (I)$$

Nous devons cependant mettre un bémol à cette première étude. En effet, lors de la résolution de (a), le polynôme  $P(\Psi)$  peut avoir des racines complexes selon les valeurs de  $t$ : nous le voyons dans la résolution du Quicksort médiane de cinq (Annexes VI et VII pages 25 à 27). Ainsi les parties réelles des termes qu'ils engendrent perturbent le terme linéaire de la solution (I). Cette perturbation doit pouvoir être quantifiée en analysant précisément les racines du polynôme  $P(\Psi)$  [Thèse de P.Hennequin, 1991]. Pour notre part, cette perturbation restant faible, nous la négligerons afin de pouvoir poursuivre notre analyse.

2<sup>ème</sup> cas :

Prenons maintenant le cas où  $t$  dépend de  $n$ , soit  $2t + 1 = \frac{n}{\beta}$  avec  $\beta \geq 1$  et  $\beta = O(1)$ .

Ainsi, nous avons l'équation :

$$C_n = \frac{3n}{2\beta} + (n+1) + \sum_{k=1}^n \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} (C_{k-1} + C_{n-k})$$

Reprenant les résultats vus dans la résolution du premier cas (Annexe X pages 31 et 32), la solution est:

$$C_n = \left( \frac{3}{2\beta} + 1 \right) \alpha (n+1) (H_{n+1} - 1) - \frac{3}{2\beta} n + K_0 (n+1) \quad (II)$$

avec  $\alpha = \frac{1}{H_{2t+2} - H_{t+1}}$  et  $\lambda = \frac{3}{2}(2t+1)$

or  $t = \frac{n}{2\beta} - \frac{1}{2}$

donc  $\alpha = \frac{1}{H_{\frac{n}{\beta}+1} - H_{\frac{n}{2\beta}+\frac{1}{2}}}$

Nous pouvons faire l'approximation suivante pour un  $n$  assez grand:

$$H_{\frac{n}{\beta}+1} \sim \ln\left(\frac{n}{\beta}\right) + \gamma + \frac{\beta}{n} + \frac{\beta}{2(n+\beta)} - \frac{\beta^2}{12(n+\beta)^2}$$

$$H_{\frac{n}{2\beta}+\frac{1}{2}} \sim \ln\left(\frac{n}{2\beta}\right) + \gamma + \frac{\beta}{n} + \frac{2\beta}{2(n+\beta)} - \frac{4\beta^2}{12(n+\beta)^2}$$

Ainsi,

$$\alpha \sim \frac{1}{\ln 2 - \frac{\beta}{2(n+\beta)} + \frac{\beta^2}{4(n+\beta)^2}}$$

$$\alpha \xrightarrow[n \text{ grand}]{\beta \text{ constant}} \frac{1}{\ln 2}$$

Après l'analyse de ces deux cas, nous pouvons maintenant décrire le comportement de l'équation (a).

- Analysons l'équation (I):

Le coefficient  $\alpha$  a une décroissance de type logarithmique avec  $t$ . En effet,  $\alpha$  tend fortement vers sa limite  $\frac{1}{\ln 2}$ :  $\alpha$  est déjà égal à  $\approx 1,491$  pour  $t = 10$ .

- Analysons l'équation (II):

Le coefficient du terme dominant est  $\alpha(\frac{3}{2\beta} + 1)$  avec  $\alpha \rightarrow \frac{1}{\ln 2}$ , son évolution est donc linéaire en fonction de  $\frac{1}{\beta}$ .

En négligeant les termes linéaires des solutions (I) et (II) (i.e  $n$  grand), nous pouvons maintenant exprimer clairement l'évolution de  $C_n$  en fonction de  $t$  par le Graphe I de l'Annexe XIX page 47.

Nous voyons sur celui-ci que, pour un  $n$  donné, le Quicksort médiane de  $2t + 1$  éléments a un nombre de comparaisons optimal pour une certaine valeur de  $2t + 1$ , que nous nommerons OPT.

OPT dépend donc de la valeur  $n$  (pour plus de détails sur ce sujet, se reporter à l'article récent [C.Martinez, S.Roura, "Optimal Sampling for Quicksort", 2000]). Et nous pouvons affirmer que pour un  $n$  donné, plus  $n$  sera grand, plus OPT sera élevé et plus  $\alpha$  sera faible. En fait, lorsque  $n$  croît, le Graphe I aura tendance à se dilater selon  $t$  tout en se creusant au niveau du point OPT.

Il faut noter que pour un  $n$  petit, cette analyse restera correcte dans la mesure où les termes linéaires de (I) et de (II) sont opposés, tous deux, à l'évolution du coefficient du terme dominant de leurs équations respectives.

Revenons maintenant à l'évolution de  $E_n$  en fonction de  $2t + 1$ . Nous l'avons représenté par le Graphe II (Annexe XIX page 48). Nous avons construit celui-ci à partir de l'approximation faite sur  $E_n$ , soit  $E_n = \alpha_e(n + 1)(H_{n+1} - 1) + O(n)$  avec  $\alpha_e = \frac{t+1}{2(2t+3)}$ . Nous pouvons voir sur celui-ci les deux extrémums:

- pour  $t = 0$ ,  $\alpha_e = \frac{1}{3}$
- pour  $t = \frac{n-1}{2}$ ,  $\alpha_e = \frac{1}{4 \ln 2}$ .

En conclusion de cette première étude, nous pouvons affirmer qu'il existe une taille d'échantillon  $2t + 1$  où les performances du Quicksort médiane de  $2t + 1$  éléments seront optimales pour un  $n$  donné. Le Graphe I nous montre que cette valeur OPT restera tout de même toujours très petite par rapport à  $n$ : nous analyserons cela de plus près dans la seconde partie. Ainsi, nous avons aussi pu constater que la valeur de OPT avait tendance à croître avec  $n$ : pour des tailles croissantes de tableau, nous aurons donc des valeurs croissantes de OPT. L'idée serait alors d'utiliser des échantillons de tailles différentes pour la recherche de la médiane selon la taille des tableaux à trier: nous commençons à voir ici l'intérêt d'une variante de Quicksort "à paliers".

Nous rejoignons aussi une méthode que l'on applique traditionnellement au Quicksort: "la coupure" que nous allons étudier dans cette seconde partie avec la coupure du Quicksort par le tri par insertion pour les tableaux de petites tailles.

## 2/Les effets bénéfiques de la méthode de la "coupure" et généralisation :

L'inconvénient majeur du Quicksort est qu'il n'est pas performant sur les tableaux de petites tailles. Ce problème a été résolu très tôt par R. Sedgwick: il préconisa de réaliser une "coupure" par le tri par insertion à partir d'une certaine valeur de  $n$ . En effet, nous savons, connaissant le nombre de comparaisons et d'échanges qu'il effectue (voir Annexe XI page 34), que le tri par insertion est un bon tri pour un  $n$  petit. De plus il est très simple à implémenter. Nous pourrions cependant faire mieux en utilisant des tris plus proches de l'optimum comme le tri par insertion dichotomique ou le tri "Merge Insertion" [Knuth, "Sorting and Searching", 1973] (voir Annexe XII page 35 pour le tri par insertion dichotomique).

En considérant  $m$  comme le point de coupure, il nous faut donc tout d'abord établir, pour chacun des paramètres, les équations de récurrence modélisant le nouveau comportement de notre algorithme. Par exemple, le nombre de comparaisons de la version (O) du Quicksort avec coupure sera solution de :

$$\begin{cases} C_n = n + 1 + \frac{1}{n} \sum_{k=1}^n (C_{k-1} + C_{n-k}) & \text{pour } n \geq m \\ C_n = \frac{n(n+3) - 4}{4} & \text{pour } n < m \end{cases}$$

Ensuite, il nous faut les résoudre: dans le 1/, nous avons vu que seules les valeurs constantes (i.e. assez petites) de  $2t + 1$  nous intéressaient pour espérer avoir un Quicksort des plus efficaces, nous allons donc

considérer dans cette deuxième étude  $t$  constant. Nous obtiendrons alors des solutions de la forme, pour un Quicksort Médiane de  $2t + 1$  éléments avec coupure par le tri par insertion:

$$\begin{cases} C_n = \alpha(n+1)H_{n+1} + f(m)(n+1) + O(1) \\ E_n = \alpha_e(n+1)H_{n+1} + f_e(m)(n+1) + O(1) \\ A_n = (n+1)f_a(m) + O(1) \\ P_n = (n+1)f_p(m) + O(1) \end{cases}$$

La démonstration de ce résultat se trouve dans l'Annexe XV page 40. Nous donnons aussi le graphe de  $m_{t_{opt}}$  (le point de coupure optimal pour  $C_n$ ) en fonction de  $t$  et le graphe de  $f(m_{t_{opt}})$  en fonction de  $t$  (voir Annexe XIX Graphe V page 51). Nous avons pu aussi tracer la courbe de  $C_n$  représentant le nombre de comparaisons en fonction de  $n$  et de  $t$  (voir Annexe XIX Graphe VI page 52) et constater ainsi que le gain en nombre de comparaisons apporté par les médianes d'un nombre plus élevé d'éléments est de plus en plus faible. En effet, la courbe  $C_n$  a une décroissance logarithmique avec  $t$ .

Nous avons donc calculé précisément ces quatre paramètres pour le Quicksort médiane de 3 et médiane de 5 (voir Annexes XIII et XIV pages 36 à 39) puis nous avons comparé l'évolution du nombre de comparaisons de ces deux algorithmes et du Quicksort médiane de 7 sur un graphique (voir Annexe XIX Graphe III page 49). Nous constatons la chose suivante: le Quicksort médiane de 3 est moins bon que le Quicksort médiane de 5 et médiane de 7, et les courbes de ces deux derniers semblent confondues. Cela nous montre que le Quicksort médiane de 5 avec coupure doit être une variante presque optimale du Quicksort, le nombre d'échanges effectués variant très faiblement d'une version à l'autre comme nous l'avons vu sur le Graphe II (Annexe XIX page 48).

Cependant, en étudiant de plus près les équations du Quicksort médiane de 5 et Quicksort médiane de 3, nous nous apercevons que ce dernier devient meilleur que le précédent pour  $n < 689$ . De la même manière, le Quicksort médiane de 7 est meilleur que le Quicksort médiane de 5 pour un  $n$  supérieur à environ  $20 * 10^6$  (voir Annexe XIX Graphe IV page 50). Nous voyons donc bien ici se dessiner notre variante du Quicksort "avec paliers", dont nous pouvons maintenant aborder l'algorithme et la complexité.

### 3/Le Quicksort à $t$ paliers, un Quicksort performant :

Nous venons de voir que le Quicksort médiane de cinq était préférable au Quicksort médiane de trois pour  $n > 689$  et le Quicksort médiane de sept au Quicksort médiane de cinq pour  $n > 20 * 10^6$ . Ainsi nous pouvons tout simplement visualiser sur un graphique (ou bien le démontrer théoriquement pour s'en persuader) que pour une variante du Quicksort médiane de  $2t + 1$ , il existe pour un  $n$  suffisamment grand, une valeur  $m_{t+1}$  pour laquelle la variante médiane de  $2t + 3$  doit effectuer moins de comparaisons. On doit donc pouvoir généraliser la méthode de la "coupure" avec les différentes variantes du Quicksort lui-même.

Nous avons par conséquent imaginé une version du Quicksort qui, au palier  $t$ , c'est-à-dire pour  $n > m_t$ , prendrait le pivot comme le médian de  $2t + 1$  éléments du tableau. Il est bien évident que le gain en temps en pratique sera très faible même pour de grandes valeurs de  $n$  ( $n \geq 10^6$ ). Cependant l'approche théorique nous semble intéressante à aborder. Nous allons donc essayé d'estimer en moyenne  $C_n$ ,  $A_n$ ,  $P_n$  pour cet algorithme afin de les comparer aux valeurs trouvées pour les versions du Quicksort précédentes.

Nous allons étudier le cas le plus compliqué, le nombre de comparaisons, l'analyse pour le nombre d'appels et de partitions en découlera.

Nous pouvons donc établir pour  $C_n$  la récurrence suivante:

paliers:

$$\begin{array}{l}
 t \quad \text{Médiane } 2t + 1 \\
 t - 1 \quad \text{Médiane } 2t - 1 \\
 \vdots \\
 \vdots \\
 1 \quad \text{Médiane de 3} \\
 0 \quad \text{Tri par insertion}
 \end{array}
 \left\{
 \begin{array}{l}
 C_n = n + 1 + \frac{3}{2}(2t + 1) + \sum_{k=1}^n \frac{C_{k-1}^t C_{n-k}^t}{C_n^{2t+1}} (C_{k-1} + C_{n-k}) \quad \text{si } n > m_t \\
 C_n = n + 1 + \frac{3}{2}(2t - 1) + \sum_{k=1}^n \frac{C_{k-1}^{t-1} C_{n-k}^{t-1}}{C_n^{2t-1}} (C_{k-1} + C_{n-k}) \quad \text{si } n > m_{t-1} \\
 \vdots \\
 \vdots \\
 C_n = n + 1 + \frac{8}{3} + \sum_{k=1}^n \frac{C_{k-1}^1 C_{n-k}^1}{C_n^3} (C_{k-1} + C_{n-k}) \quad \text{si } n > m_1 \\
 C_n = \frac{n(n+3) - 4}{4} \quad \text{sinon}
 \end{array}
 \right.$$

Nous n'utiliserons pas ici la version (O) du Quicksort car elle est moins performante que le Quicksort médiane de trois quel que soit  $n$ .

La résolution de cette récurrence se fait sur le même modèle que les résolutions de la partie précédente.

Sachant que  $t$  doit rester constant par rapport à  $n$  ( $t = O(1)$ ):

$$\begin{aligned}
 \frac{C_n}{n+1} &= \alpha_t \sum_{i=m_t+2}^{n+1} \frac{1}{i} + \alpha_{t-1} \sum_{i=m_{t-1}+2}^{m_t+1} \frac{1}{i} + \dots + \alpha_1 \sum_{i=m_1+2}^{m_2+1} \frac{1}{i} + \lambda_t \sum_{i=m_t+1}^n \frac{1}{i(i+1)} + \dots \\
 &\dots + \lambda_{t-1} \sum_{i=m_{t-1}+1}^{m_t} \frac{1}{i(i+1)} + \dots + \lambda_1 \sum_{i=m_1+1}^{m_2} \frac{1}{i(i+1)} + \frac{C_{m_1}^*}{m_1+1}
 \end{aligned}$$

La solution est alors de la forme:

$$C_n = \alpha_t(n+1)H_{n+1} + \left( \sum_{i=2}^t f_i(m_{i_{opt}}) + f_1(m_{1_{opt}}) \right) (n+1) - \lambda_t$$

avec

$$\begin{cases}
 f_t(m_t) = (\alpha_{t-1} - \alpha_t)H_{m_t+1} + \frac{\lambda_t - \lambda_{t-1}}{m_t + 1} \\
 \vdots \\
 f_2(m_2) = (\alpha_1 - \alpha_2)H_{m_2+1} + \frac{\lambda_2 - \lambda_1}{m_2 + 1}
 \end{cases}$$

et  $f_1(m_1)$  est particulière puisque c'est la coupure par le tri par insertion:

$$f_1(m_1) = -\alpha_1 H_{m_1+1} + \frac{\lambda_1}{m_1 + 1} + \frac{C_{m_1}^*}{m_1 + 1}$$

On peut exprimer  $f'_t(m_t)$  pour  $t \neq 1$ :

$$f'_t(m_t) = \frac{\alpha_{t-1} - \alpha_t}{m_t + 1} - \frac{\lambda_t - \lambda_{t-1}}{(m_t + 1)^2}$$

$$f'_t(m_t) = \frac{(\alpha_{t-1} - \alpha_t)(m_t + 1) - (\lambda_t - \lambda_{t-1})}{(m_t + 1)^2}$$

Résoudre  $f'_t(m_t) = 0$  nous permet alors de trouver le minimum de  $f_t$ :

pour  $t > 1$ , 
$$m_{t_{opt}} = \frac{\lambda_t - \lambda_{t-1}}{\alpha_{t-1} - \alpha_t} - 1$$

et  $m_{1_{opt}}$  est donné dans la partie précédente (coupure entre le Quicksort Médiane de 3 ( $t = 1$ ) et le tri par insertion). Par substitution, nous obtenons:

$$f_t(m_{t_{opt}}) = (\alpha_{t-1} - \alpha_t) \left( H_{\frac{\lambda_t - \lambda_{t-1}}{\alpha_{t-1} - \alpha_t}} + 1 \right)$$

En prenant l'hypothèse  $\lambda_t = \frac{3}{2}(2t + 1)$ , nous obtenons  $\lambda_t - \lambda_{t-1} = 3$  et par conséquent:

$$m_{t_{opt}} = \frac{3}{\alpha_{t-1} - \alpha_t} - 1$$

et

$$f_t(m_{t_{opt}}) = (\alpha_{t-1} - \alpha_t) \left( H_{\frac{3}{\alpha_{t-1} - \alpha_t}} + 1 \right)$$

Nous donnons le graphe de  $m_{t_{opt}}$  en fonction de  $t$  et le graphe de  $f_t(m_{t_{opt}})$  en fonction de  $t$  pour  $t > 1$  (voir Annexe XIX Graphe VII page 53); nous y voyons l'évolution des points de coupure de notre Quicksort à  $t + 1$  paliers. Nous avons pu aussi tracer la courbe de  $f_t(m_{t_{opt}})$  cumulée en fonction de  $t$  et  $C_n$  représentant le nombre de comparaisons en fonction de  $n$  et de  $t$  (voir Annexe XIX Graphe VIII page 54) et constater ainsi que le gain en nombre de comparaisons apporté par les médianes d'un nombre plus élevé d'éléments est de plus en plus faible. En effet, tout comme dans le 2/, la courbe  $C_n$  a une décroissance logarithmique avec  $t$ .

Cependant, la somme composant le terme linéaire mérite une analyse toute particulière car c'est elle qui devrait permettre au Quicksort "à  $t$  paliers" d'effectuer un nombre de comparaisons moins élevé que le Quicksort médiane de  $2t + 1$  éléments. Mais qu'en est-il donc ?

Nous avons donc comparé l'évolution du coefficient du terme linéaire de ces deux variantes (voir le second graphe Annexe XIX Graphe VI page 50). Nous pouvons y voir que le coefficient du terme linéaire du Quicksort à  $t$  paliers a une croissance très faible en comparaison avec ceux des versions précédentes du Quicksort. En fait, il converge selon  $t$  (voir le premier graphe page 14). En effet, on peut vérifier que:

$$\lim_{t \rightarrow +\infty} f_t(m_{t_{opt}}) = 0$$

Par conséquent,

$$\sum_{i=2}^t f_i(m_{i_{opt}}) \text{ converge}$$

Nous n'aurons certainement pas ce résultat en tenant compte de la perturbation que nous évoquions plus haut. Cependant l'approximation que nous avons faite se répercute sur les trois versions du Quicksort. Ainsi, nous pouvons affirmer que:

Pour un  $t$  fixé, le Quicksort à paliers effectuera un nombre de comparaisons inférieur au Quicksort médiane de  $2t + 1$  avec coupure par le tri par insertion.

Nous n'avons pas fait le calcul pour le nombre d'échanges mais en revanche, en suivant la même méthode de résolution, nous avons pu démontrer que, pour un  $t$  fixé, les nombres d'appels  $A_n$  et de partitions  $P_n$  étaient identiques à ceux du Quicksort médiane de  $2t + 1$  éléments avec coupure par le tri par insertion.

Pour conclure ce premier chapitre, nous donnons en exemple les résultats de cette méthode pour un nombre de paliers  $t = 3$  (voir Annexes XVI et XVII pages 42 à 44).

Nous pensons que cette version, d'après nos expérimentations sur les performances en temps machine de celle-ci, le nombre d'échanges augmentant, est presque optimale pour des tailles de tableaux de l'ordre du millions d'éléments. En effet, le gain en temps apporté par des versions plus élaborées n'est que de quelques millisecondes, même pour des  $n$  très grands ( $10^6$ ). Cependant, ce gain reste bien évidemment croissant avec  $n$ , c'est pourquoi, l'utilisation d'un algorithme à 5 ou 6 paliers peut être tout de même intéressant pour des valeur de  $n$  dépassant la centaine de millions d'éléments.

Même une méthode de recherche d'un médian approché (bien plus rapide que la recherche d'un médian exact), ne nous donnent pas des résultats vraiment meilleurs en pratique. Son approche théorique reste cependant intéressante à étudier pour évaluer ces performances avec plus de précision (comme nous l'avons dit ci-dessus, elle peut s'avérer utile pour des  $n$  très grand). Celle-ci se mène de la même manière qu'avec un médian exact, seules les probabilités associées à la position  $k$  du pivot changent en fonction du pourcentage d'erreur induit par l'approximation du médian. Nous en voyons un exemple dans l'Annexe XVII page 44 avec un Quicksort "médiane approchée de 9 éléments".

Nous allons maintenant aborder le Chapitre II: l'étude de la variante du Quicksort en  $O(n \ln n)$  dans le pire des cas, où nous utiliserons des raisonnements des résultats du Chapitre I.

## CHAPITRE II: QUICKSORT EN $O(n \log_2 n)$ DANS LE PIRE DES CAS

Pour que le Quicksort s'exécute en  $O(n \log_2 n)$ , il faut tout simplement que la position du pivot soit telle, qu'à chaque appel, elle divise notre tableau de taille  $n$  en deux sous-tableaux de tailles  $n_1$  et  $n_2$  de telle manière à ce que  $n_1$  et  $n_2$  soient fonctions de  $n$ .

Voilà un exemple de découpage vérifiant cette hypothèse:

$$T_n = O(n) + T_{\lfloor \frac{1}{\alpha} \rfloor} + T_{\lceil \frac{\alpha}{10} n \rceil}$$

L'idée serait donc de prendre le pivot comme le médian d'un échantillon de taille  $2t + 1 = f(n)$ . Mais il faut alors que la recherche de ce médian s'effectue en  $O(n)$  comparaisons dans le pire des cas pour que l'algorithme final s'exécute en  $O(n \log_2 n)$ .

Prenons donc  $M_n$  le nombre de comparaisons effectué par la recherche de la médiane dans le pire des cas et choisissons  $f(n) = \lfloor \frac{n}{\beta} \rfloor$  avec  $\beta = O(1)$ . La récurrence pour le nombre de comparaisons du Quicksort dans le pire des cas deviendrait alors:

$$C_n \leq M_{\lfloor \frac{n}{\beta} \rfloor} + n - 1 + C_{\lfloor \frac{\lfloor \frac{n}{\beta} \rfloor}{2} \rfloor} + C_{n - \lfloor \frac{\lfloor \frac{n}{\beta} \rfloor}{2} \rfloor}$$

$$C_n \leq M_{\frac{n}{\beta}} + n + C_{\frac{n}{2\beta}} + C_{n - \frac{n}{2\beta}}$$

$$C_n \leq M_{\frac{n}{\beta}} + n + C_{\frac{n}{2\beta}} + C_{\frac{(2\beta-1)n}{2\beta}}$$

Nous allons résoudre cette équation en prenant donc comme hypothèse  $C_n \leq \alpha n \ln n$ .

Cela donne:

$$C_n \leq M_{\frac{n}{\beta}} + n + \frac{\alpha}{2\beta} n \ln \left( \frac{n}{2\beta} \right) + \alpha \left( \frac{2\beta-1}{2\beta} \right) n \ln \left( \frac{2\beta-1}{2\beta} n \right)$$

$$C_n \leq \alpha n \ln n + n \left( 1 + \alpha \frac{(2\beta-1) \ln(2\beta-1) - 2\beta \ln(2\beta)}{2\beta} \right) + M_{\frac{n}{\beta}}$$

En prenant  $M_n \leq an$ , on obtient:

$$C_n \leq \alpha n \ln n + n \left( 1 + \frac{a}{\beta} + \alpha \frac{(2\beta-1) \ln(2\beta-1) - 2\beta \ln(2\beta)}{2\beta} \right)$$

Pour vérifier l'hypothèse, il nous faut:

$$1 + \frac{a}{\beta} + \alpha \frac{(2\beta-1) \ln(2\beta-1) - 2\beta \ln(2\beta)}{2\beta} \leq 0$$

soit

$$\alpha \geq \frac{2(a+\beta)}{2\beta \ln(2\beta) - (2\beta-1) \ln(2\beta-1)} \quad (d)$$

Et nous pouvons vérifier que cette relation permet de respecter les conditions initiales de la récurrence.

Nous pouvons remarquer que, pour  $\beta = 1$ , nous obtenons:

$$\alpha \geq \frac{a+1}{\ln 2}$$

Connaissant la valeur  $a$ , nous pourrions calculer la valeur de  $\alpha$  minimale en fonction de  $\beta$ .

Pour cela, nous avons étudié une variante du célèbre algorithme de sélection en temps linéaire dans le pire des cas de R.Rivest et R.Tarjan (1973). Le nombre de comparaisons qu'il effectue, noté  $S_n$ , est

$$M_n \leq 12n$$

Mais cette borne est évidemment bien trop élevée pour nous donner une valeur de  $\alpha$  intéressante.

Dernièrement une borne supérieure intéressante a été donnée pour  $S_n$  par D.Dor et U.Zwick ["Selecting the median", 1999]

$$M_n \leq 3n$$

C'est-à-dire  $a \leq 3$ .

La relation (d) devient donc

$$\alpha \geq \frac{6+2\beta}{2\beta \ln(2\beta) - (2\beta-1) \ln(2\beta-1)}$$

Nous donnons le graphe de  $\alpha$  en fonction de  $\beta$  (voir le second graphe à la page 14).

Le minimum de cette fonction,  $\alpha_{min}$ , s'obtient pour  $\beta$  solution de l'équation  $7 \ln(2\beta-1) - 6 \ln 2\beta = 0$ .

Ce qui nous donne:

$$\alpha_{min} = 4.3962 \quad \text{pour} \quad \beta = 2.4575$$

Pour être tout à fait complet, il nous faut traiter plus précisément le cas où  $\beta = 1$ . En effet, l'algorithme de recherche du médian partitionne aussi les valeurs du tableau autour de ce même médian. C'est pourquoi dans le cas  $\beta = 1$ ,  $M_n$  nous donne le nombre de comparaisons nécessaire pour trouver le médian et partitionner les éléments du tableau autour de celui-ci.

Pour  $\beta = 1$ , la récurrence s'écrit donc

$$C_n \leq M_n + C_{\lfloor \frac{n}{2} \rfloor} + C_{n - \lceil \frac{n}{2} \rceil}$$

Toujours en prenant l'hypothèse  $C_n \leq \alpha n \ln n$ , nous aurons cette fois:

$$\alpha \geq \frac{a}{\ln 2}$$

Avec  $a = 3$ , cela nous donne

$$\alpha_{min} = \frac{3}{\ln 2}$$

Pour  $\beta = 1$ , nous trouvons donc une borne légèrement meilleure que précédemment soit  $\alpha_{min} = 4.3281$ .

Finalement, la meilleure borne pour le nombre de comparaisons que nous avons pu obtenir pour le Quicksort dans le pire des cas est donc  $C_n = 3n \log_2 n$ . Celui-ci ne semble donc pas pouvoir rivaliser avec un tri comme le Heapsort qui lui effectue  $C_n = 2n \log_2 n$  comparaisons dans le pire des cas.

Cependant, en utilisant des méthodes plus élaborées, notamment des algorithmes de recherche approximative du médian et non pas exacte, nous pourrions peut-être encore abaisser cette borne. Cette étude nécessiterait donc de plus amples recherches.

Pour conclure ce chapitre, nous pouvons maintenant faire une remarque essentielle sur le comportement du Quicksort.

Nous avons défini un algorithme du Quicksort dans le pire des cas, il effectue donc

$$C_n \leq 3n \log_2 n \quad (\beta = 1)$$

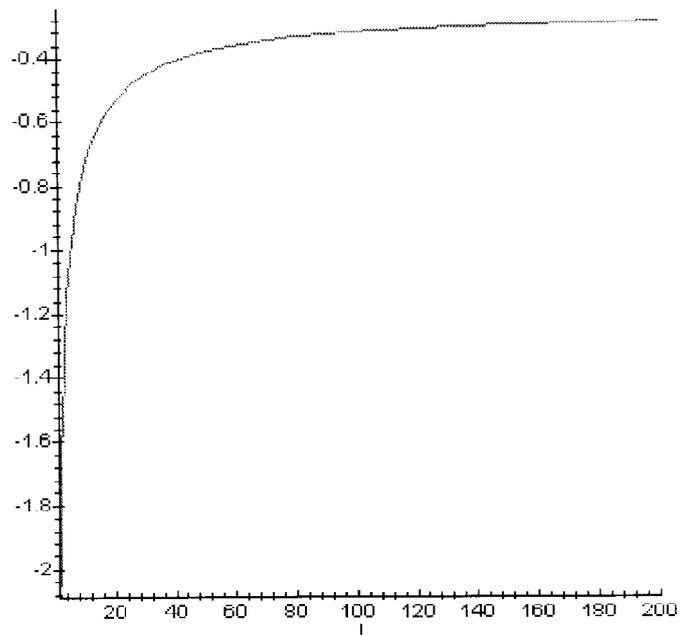
Revenons à l'équation (II) page 6 et regardons le nombre de comparaisons qu'effectue en moyenne cet algorithme, si la recherche du médian s'effectue en moyenne en au moins  $\frac{3}{2}n$ :

$$\overline{C}_n \geq 2.5n \log_2 n$$

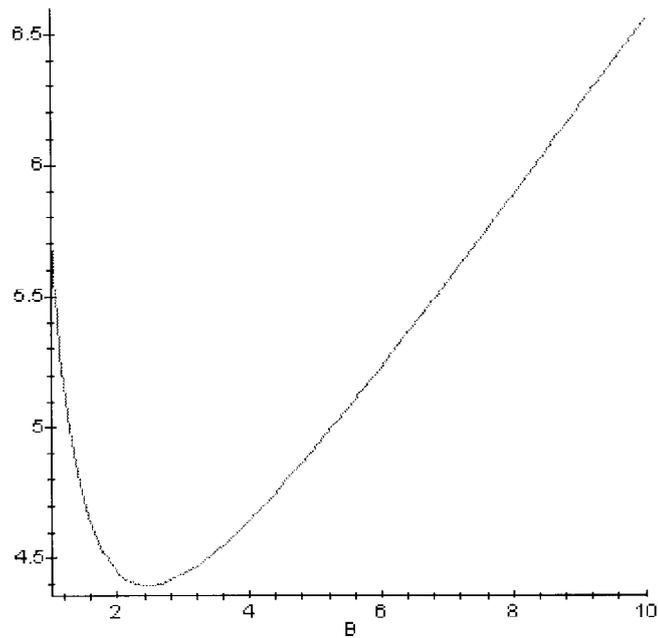
Les performances en moyenne de notre algorithme optimal dans le pire des cas se détériore nettement. Inversement, un algorithme du Quicksort efficace en moyenne c'est-à-dire avec un échantillon de taille  $O(1)$  pour la recherche du médian, s'exécutera en  $O(n^2)$  dans le pire des cas.

Nous pouvons donc établir une proposition sur "le dilemme cas moyen - cas pire" pour l'algorithme du Quicksort:

**Du point de vue du nombre de comparaisons, un gain en efficacité pour le Quicksort dans le pire des cas se traduira par une perte d'efficacité en moyenne. Et inversement, un gain en efficacité en moyenne pour le Quicksort se traduira par une perte d'efficacité dans le pire des cas.**



**Grphe du coefficient linéaire du Quicksort à t paliers en fonction de t**



**Grphe de  $\alpha$  (le coefficient du terme dominant du Quicksort en  $n \ln n$  dans le pire des cas) en fonction de  $\beta$**

## CONCLUSION

Ce travail de recherche nous a amené à établir des résultats généraux sur les variantes de Quicksort médiane de  $2t + 1$  éléments avec et sans coupure, ce qui nous a permis de mieux cerner leur comportement. De ces deux études découle naturellement l'optimisation Quicksort "à  $t$  paliers", que nous avons analysé théoriquement et pratiquement, et dont nous pouvons affirmer qu'elle est plus performante que les deux précédentes.

Nous préconisons donc l'emploi de la version "Quicksort à trois paliers", que nous avons étudiée précisément comme exemple du Quicksort "à  $t$  paliers" (Annexe XVI pages 42 et 43).

Celle-ci semble être pour des tailles de tableaux raisonnables une procédure de tri efficace, le pire des cas ne se produisant que pour des répartitions des données très particulières, et donc statistiquement peu probables.

Mais l'idée force de ce travail de recherche est que, la quantité des données à trier augmentant considérablement avec l'essor des nouvelles technologies, nous ne pouvons plus ne faire confiance qu'à l'évolution technique de l'architecture des machines, mais bien se reconcentrer sur l'optimisation algorithmique des procédures de tri.

## ANNEXES

ANNEXE I	
Exemple de calcul de la hauteur de pile . . . . .	p.18
ANNEXE II	
Nombre de comparaisons pour QS Médiane de 3 . . . . .	p.20
ANNEXE III	
Nombre d'échanges pour QS Médiane de 3 . . . . .	p.22
ANNEXE IV	
Nombre d'appels pour QS Médiane de 3 . . . . .	p.23
ANNEXE V	
Nombre de partitions pour QS Médiane de 3 . . . . .	p.24
ANNEXE VI	
Nombre de comparaisons pour QS Médiane de 5 . . . . .	p.25
ANNEXE VII	
Nombre d'échanges pour QS Médiane de 5 . . . . .	p.27
ANNEXE VIII	
Nombre d'appels pour QS Médiane de 5 . . . . .	p.28
ANNEXE IX	
Nombre de partitions pour QS Médiane de 5 . . . . .	p.29

ANNEXE X	
Nombre de comparaisons pour QS Médiane de $2t+1$ . . . . .	p.30
ANNEXE XI	
Tri par insertion . . . . .	p.32
ANNEXE XII	
Tri par insertion dichotomique . . . . .	p.33
ANNEXE XIII	
Calcul du point de coupure pour QS Médiane de 3 . . . . .	p.34
ANNEXE XIV	
Calcul du point de coupure pour QS Médiane de 5 . . . . .	p.36
ANNEXE XV	
Nombre de comparaisons pour QS Médiane de $2t+1$ avec coupure à $m$ . . . . .	p.38
ANNEXE XVI	
Quicksort à trois paliers . . . . .	p.40
ANNEXE XVII	
Quicksort avec Médiane de 9 approché + Résultats pratiques . . . . .	p.42
ANNEXE XVIII	
Tableaux de résultats . . . . .	p.43
ANNEXE XIX	
Graphiques . . . . .	p.45

ANNEXE I: Exemple de calcul de hauteur de pile

Pour les différentes variantes que nous analyserons, nous ne recalculerons pas  $Hp_n$ , qui est dans le meilleur des cas  $\log_2 n$ , dans le pire des cas  $n$ . Nous calculons donc ici  $Hp_n$  dans le cas moyen pour la version (O) du Quicksort présenté dans l'introduction:

Etablissons tout d'abord une borne supérieure:

$$Hp_n \leq 1 + \frac{1}{n} \sum_{k=1}^n \max(Hp_{n-k}, Hp_{k-1})$$

$$Hp_n \leq 1 + \frac{1}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} Hp_k$$

$$Hp_n \leq 1 + \frac{1}{n} \left( \sum_{k=1}^{n-1} Hp_k - \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor - 1} Hp_k \right)$$

Prenons alors  $Hp_n \leq c \ln n$

$$Hp_n \leq 1 + \frac{1}{n} \left( \sum_{k=1}^{n-1} c \ln k - \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor - 1} c \ln k \right)$$

$$Hp_n \leq 1 + \frac{c}{n} \left( \sum_{k=1}^{n-1} \ln k - \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor - 1} \ln k \right)$$

or

$$\sum_{k=1}^n \ln k \leq (n+1) \ln(n+1) - (n+1)$$

donc

$$Hp_n \leq 1 + \frac{c}{n} \left( n \ln n - n - \lfloor \frac{n}{2} \rfloor \ln \left( \lfloor \frac{n}{2} \rfloor \right) + \lfloor \frac{n}{2} \rfloor \right)$$

$$Hp_n \leq 1 + \frac{c}{n} \left( n \ln n - n + \lfloor \frac{n}{2} \rfloor \right)$$

$$Hp_n \leq 1 + \frac{c}{n} \left( n \ln n - \frac{n}{2} \right)$$

$$Hp_n \leq c \ln n - \frac{c}{2} + 1$$

Les conditions initiales sont: 
$$\begin{cases} Hp_0 \leq 1 \\ Hp_1 \leq 2 \\ Hp_2 \leq 3 \\ Hp_3 \leq 4 \end{cases}$$

Pour les satisfaire, on peut prendre  $c = \frac{2}{\ln 2}$ :

$$Hp_n \leq \frac{2}{\ln 2} \ln n$$

donc

$$Hp_n \leq 2 \log_2 n$$

Par exemple:

théoriquement:  $Hp_{1000} = 19.9$  et  $Hp_{1000000} = 39.8$   
expérimentalement:  $Hp_{1000} = 19$  et  $Hp_{1000000} = 38$

ANNEXE II: Calcul théorique du nombre de comparaisons Médiane de 3

Le nombre de comparaisons en moyenne est donné par la récurrence suivante :

$$C_n = n + 1 + \frac{8}{3} + \sum_{k=1}^n \frac{C_{k-1}^1 C_{n-k}^1}{C_n^3} (C_{n-1} + C_{n-k})$$

avec:  $C_0 = 0, \quad C_1 = 0, \quad C_2 = 3$

$$n(n-1)(n-2)C_n = (n+1 + \frac{8}{3})n(n-1)(n-2) + 12 \sum_{k=1}^n (n-k)(k-1)C_{k-1}$$

Soit

$$C(z) = \sum_{n=0}^{+\infty} C_n z^n$$

$$\sum_{n=3}^{+\infty} n(n-1)(n-2)C_n z^{n-3} = \sum_{n=3}^{+\infty} (n+1 + \frac{8}{3})n(n-1)(n-2)z^{n-3} + 12 \sum_{n=3}^{+\infty} \sum_{k=0}^n (n-k)(k-1)C_{k-1} z^{n-3}$$

Le dernier terme est le produit de Cauchy de deux séries simples:

$$\sum_{n=3}^{+\infty} \sum_{k=0}^n (n-k)(k-1)C_{k-1} z^{n-3} = \sum_{n=0}^{+\infty} n z^{n-1} \sum_{n=0}^{+\infty} n C_n z^{n-1} = \frac{C'(z)}{(1-z)^2}$$

On obtient l'équation différentielle:

$$C'''(z) = \frac{24}{(1-z)^5} + \frac{16}{(1-z)^4} + 12 \frac{C'(z)}{(1-z)^2}$$

$$(1-z)^3 C'''(z) = \frac{24}{(1-z)^2} + \frac{16}{1-z} + 12(1-z)C'(z)$$

Notons  $\Psi(\Psi+1)\dots(\Psi+t-1) \equiv (1-z)^t C^{(t)}(z)$

L'équation d'Euler devient:

$$\Psi(\Psi+1)(\Psi+2)C(z) - 12\Psi C(z) = \frac{24}{(1-z)^2} + \frac{16}{(1-z)}$$

$$\Psi(\Psi+5)(\Psi-2)C(z) = \frac{24}{(1-z)^2} + \frac{16}{(1-z)}$$

Posons:

$$\Psi U(z) = \frac{24}{(1-z)^2} + \frac{16}{(1-z)}$$

$$(\Psi + 5)T(z) = U(z)$$

$$(\Psi - 2)C(z) = T(z) \quad (1)$$

Les conditions initiales sont alors:

$$C(0) = 0 \quad T(0) = 0 \quad U(0) = 6$$

Cela donne:

$$U(z) = \frac{12}{(1-z)^2} + \frac{16}{(1-z)} - 22$$

$$T(z) = \frac{12}{7} \frac{1}{(1-z)^2} + \frac{8}{3} \frac{1}{(1-z)} - \frac{22}{5} + \frac{2}{105} (1-z)^5 \quad (2)$$

$$C(z) = -\frac{12}{7} \frac{1}{(1-z)^2} \ln(1-z) + \frac{8}{3} \frac{z}{(1-z)^2} + \frac{11}{5} - \frac{2}{735} (1-z)^5 - \frac{323}{147} \frac{1}{(1-z)^2}$$

De là on obtient:

$$C_n = \frac{12}{7}(n+1)H_{n+1} + \frac{23}{49}n + \frac{2}{735}$$

Soit

$$C_n = \frac{12}{7}(n+1) \ln(n+1) - 0.255n$$

ANNEXE III: Calcul théorique du nombre d'échanges Médiante de 3

Le nombre d'échanges en moyenne lors du partitionnement est donné par:

$$Ep_n = \sum_{k=1}^n \frac{C_{k-1}^1 C_{n-k}^1}{C_n^3} \frac{(n-k)(k-1)}{n-1}$$

$$Ep_n = \frac{n^2 - 2n + 2}{5(n-1)}$$

Donc le nombre d'échanges est donné par la récurrence suivante:

$$E_n = Ep_n + 2 + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (E_{n-k} + E_{k-1})$$

$$E_n = \frac{n^2 + 8n - 8}{5(n-1)} + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (E_{n-k} + E_{k-1})$$

avec: 
$$\begin{cases} E_0 = 0 \\ E_1 = 0 \\ E_2 = 1 \end{cases}$$

$$E(z) = \sum_{n=0}^{+\infty} E_n z^n$$

On arrive à:

$$E'''(z) = \frac{24}{5} \frac{1}{(1-z)^5} + \frac{48}{5} \frac{1}{(1-z)^4} + \frac{2}{5} \frac{1}{z(1-z)^3} - \frac{1}{5} \frac{1}{z^2(1-z)^2} + \frac{1}{5z^2} + 12 \frac{E'(z)}{(1-z)^2}$$

$$\Psi(\Psi + 1)(\Psi + 2)E(z) - 12\Psi E(z) = \frac{24}{5} \frac{1}{(1-z)^2} + \frac{48}{5} \frac{1}{1-z} + \frac{2}{5z} - \frac{1}{5} \frac{(1-z)}{z^2} + \frac{(1-z)^3}{5z^2}$$

En résolvant comme lors du calcul du nombre de comparaisons Médiante de 3, nous obtenons avec

les conditions initiales  $U(0) = 2, T(0) = 0, E(0) = 0$ :

$$U(z) = \frac{12}{5} \frac{1}{(1-z)^2} + \frac{48}{5} \frac{1}{1-z} - \frac{2}{5} \ln(1-z) + \frac{2}{5} - 10$$

$$T(z) = \frac{12}{35} \frac{1}{(1-z)^2} + \frac{24}{15} \frac{1}{1-z} - \frac{1013}{500} - \frac{2}{25} \ln(1-z) + \frac{1}{20} z + \frac{291}{3500} (1-z)^5$$

$$E(z) = -\frac{12}{35} \frac{\ln(1-z)}{(1-z)^2} - \frac{213}{500} \frac{z}{(1-z)^2} + \frac{519}{500} \frac{z^2}{(1-z)^2} - \frac{1}{60} \frac{z^3}{(1-z)^2} - \frac{1}{50} + \frac{781}{24500} \frac{1}{(1-z)^2}$$

De là, on tire

$$E_n = \frac{12}{35}(n+1)H_{n+1} + \frac{209}{735}n - \frac{967}{735}$$

Soit

$$E_n = \frac{12}{35}(n+1) \ln(n+1) + 0.482n$$

ANNEXE IV: Calcul théorique du nombre d'appels pour la médiane de 3

Le nombre d'appels est donné par la récurrence:

$$A_n = 1 + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (A_{k-1} + A_{n-k})$$

avec  $\begin{cases} A_0 = 1 \\ A_1 = 1 \\ A_2 = 3 \end{cases}$

En posant

$$A(z) = \sum_{n=0}^{+\infty} A_n z^n \quad ,$$

on obtient:

$$A'''(z) = \frac{6}{(1-z)^4} + 12 \frac{A'(z)}{1-z}$$

Le schéma de résolution étant toujours le même, cela nous donne:

avec  $\begin{cases} U(0) = -2 \\ T(0) = -1 \\ A(0) = 1 \end{cases}$

$$U(z) = \frac{6}{1-z} - 8$$

$$T(z) = \frac{1}{1-z} - \frac{8}{5} - \frac{2}{5}(1-z)^5$$

$$\text{et } A(z) = \frac{z}{(1-z)^2} - \frac{4}{5} + \frac{1}{7} \frac{1}{(1-z)^2} - \frac{2}{35}(1-z)^5$$

On en tire:

$$A_n = \frac{8}{7}n + \frac{1}{7}$$

ANNEXE V: Calcul théorique du nombre de partitions pour la Médiane de 3

Le nombre de partitions est donné par la récurrence:

$$P_n = 1 + \sum_{k=1}^n \frac{(n-k)(k-1)}{C_n^3} (P_{k-1} + P_{n-k})$$

$$\text{où } \begin{cases} P_0 = 0 \\ P_1 = 0 \\ P_2 = 1 \end{cases}$$

En posant

$$P(z) = \sum_{n=0}^{+\infty} P_n z^n \quad ,$$

on obtient:

$$P'''(z) = \frac{6}{(1-z)^4} + 12 \frac{P'(z)}{1-z}$$

Ce qui, en suivant les schémas de résolution précédents, donne:

$$\text{avec } \begin{cases} U(0) = 2 \\ T(0) = 0 \\ P(0) = 0 \end{cases}$$

$$U(z) = \frac{6}{1-z} - 4$$

$$T(z) = \frac{1}{1-z} - \frac{4}{5} - \frac{1}{5}(1-z)^5$$

$$P(z) = \frac{z}{(1-z)^2} + \frac{2}{5} - \frac{3}{7} \frac{1}{(1-z)^2} + \frac{1}{35}(1-z)^5$$

Donc

$$P_n = \frac{4}{7}n - \frac{3}{7}$$

Le nombre de comparaisons en moyenne est donné par la récurrence suivante :

$$C_n = n + 1 + 6 + \sum_{k=1}^n \frac{C_{k-1}^2 C_{n-k}^2}{C_n^5} (C_{n-k} + C_{k-1})$$

$$\text{avec } \begin{cases} C_0 = 0 \\ C_1 = 0 \\ C_2 = 3 \\ C_3 = 6 \\ C_4 = \frac{19}{2} \end{cases}$$

$$n(n-1)(n-2)(n-3)(n-4)C_n = (n+7)n(n-1)(n-2)(n-3)(n-4) + 60 \sum_{k=1}^n (n-k)(n-k-1)(k-1)(k-2)C_{k-1}$$

Posons  $C(z) = \sum_{n=0}^{+\infty} C_n z^n$

Multiplions l'équation par  $z^{n-5}$  et sommons sur n de 5 à  $+\infty$  :

$$(1-z)^5 C^{(5)}(z) = \frac{720}{(1-z)^2} + \frac{720}{(1-z)} + 120(1-z)^2 C''(z)$$

Posons  $C''(z) = U(z)$  et l'équation d'Euler devient:

$$(\Psi(\Psi+1)(\Psi+2) - 120)U(z) = \frac{720}{(1-z)^4} + \frac{720}{(1-z)^3}$$

$$(\Psi-4)(\Psi-X_1)(\Psi-X_2)D(z) = \frac{720}{(1-z)^4} + \frac{720}{(1-z)^3}$$

avec  $X_1 = -\frac{7}{2} - \frac{1}{2}i\sqrt{71}$  et  $X_2 = -\frac{7}{2} + \frac{1}{2}i\sqrt{71}$

Posons:

$$(\Psi - X_1)S(z) = \frac{720}{(1-z)^4} + \frac{720}{(1-z)^3}$$

$$(\Psi - X_2)T(z) = S(z)$$

$$(\Psi - 4)U(z) = T(z)$$

$$\text{avec } \begin{cases} S(0) = 90 - 6i\sqrt{71} \\ T(0) = 12 \\ U(0) = 6 \end{cases}$$

En résolvant ces trois équations différentielles, on obtient:

$$\begin{cases} S(z) = \frac{222i\sqrt{71}z - 2886z - 402i\sqrt{71} + 5586}{37(1-z)^4} + \left(\frac{180}{37}i\sqrt{71} - \frac{2256}{37}\right)(1-z)^{\frac{1}{2}i\sqrt{71} + \frac{1}{2}} \\ T(z) = \frac{360}{37} \frac{1}{(1-z)^4} + 12 \frac{1}{(1-z)^3} + \left(-\frac{180}{37} - \frac{2256}{2627}i\sqrt{71}\right)(1-z)^{\frac{1}{2}i\sqrt{71} + \frac{7}{2}} + \left(-\frac{180}{37} + \frac{2256}{2627}i\sqrt{71}\right)(1-z)^{\frac{7}{2} - \frac{1}{2}i\sqrt{71}} \\ U(z) = -\frac{360 \ln(1-z)}{37} \frac{1}{(1-z)^4} + 12 \frac{z}{(1-z)^4} + \frac{5736}{1369} \frac{1}{(1-z)^4} \end{cases}$$

Par conséquent, avec  $C(0) = 0$  et  $C'(0) = 0$ :

$$C(z) = -\frac{60 \ln(1-z)}{37} \frac{1}{(1-z)^2} - \frac{50}{37} \frac{1}{(1-z)^2} + \frac{6z-4}{(1-z)^2} + \frac{5736}{1369} \frac{1}{6(1-z)^2}$$

Ainsi:

$$C_n = \frac{60}{37}(n+1)H_{n+1} - \frac{376}{1369}n + \frac{8590}{1369}$$

Soit

$$C_n = \frac{60}{37}(n+1) \ln(n+1) + 0.661n$$

**ANNEXE VII: Calcul théorique du nombre d'échanges Médiane de 5**

Le nombre d'échanges dans la partition est:

$$Ep_n = \sum_{k=1}^n \frac{C_{k-1}^2 C_{n-k}^2 (n-k)(k-1)}{C_n^5 (n-1)}$$

$$Ep_n = \frac{3n^2 + 23n - 22}{14(n-1)}$$

Le nombre d'échanges est donc donné par la récurrence:

$$E_n = 2 + Ep_n + \sum_{k=1}^n \frac{C_{k-1}^2 C_{n-k}^2}{C_n^5} (E_{n-k} + E_{k-1})$$

soit

$$E_n = \frac{3n^2 + 23n - 22}{14(n-1)} + \sum_{k=1}^n \frac{C_{k-1}^2 C_{n-k}^2}{C_n^5} (E_{n-k} + E_{k-1})$$

En suivant le même raisonnement de calcul que celui effectué lors du calcul du nombre de comparaisons Médiane de 5, on obtient les équations différentielles suivantes:

$$S(z) = \frac{13727z^2 - 231657z - 1998z^3 - 27354i\sqrt{71} + 18241i\sqrt{71}z - 1369i\sqrt{71}z^2 + 222i\sqrt{71}z^3 + 373828}{9842(1-z)^4}$$

$$+ \left( \frac{12591}{9842} i\sqrt{71} - \frac{23175}{1406} \right) (1-z)^{\frac{1}{2}i\sqrt{71} + \frac{7}{2}}$$

$$T(z) = \frac{1369z^2 - 222z^3 - 18241z + 27354}{4921(1-z)^4} - \left( \frac{162225i\sqrt{71} + 893961}{698782} \right) (1-z)^{\frac{7}{2} + \frac{1}{2}i\sqrt{71}} + \left( \frac{23175}{99826} i\sqrt{71} - \frac{12591}{9842} \right) (1-z)^{\frac{7}{2} - \frac{1}{2}i\sqrt{71}}$$

$$U(z) = \left( \frac{552763}{364154} + \frac{66}{19}z - \frac{31}{266}z^2 + \frac{2}{133}z^3 - \frac{540}{259} \ln(1-z) \right) \frac{1}{(1-z)^4}$$

Les conditions initiales étant: 
$$\begin{cases} S(0) = \frac{43}{2} - \frac{3}{2}i\sqrt{71} \\ T(0) = 3 \\ U(0) = 2 \end{cases}$$

En intégrant deux fois U(z), on obtient donc:

$$E(z) = -\frac{90}{259} \frac{\ln(1-z)}{(1-z)^2} - \frac{75}{259} \frac{1}{(1-z)^2} + \frac{-6z^3 + 12z^2 + 3z - 8}{399(1-z)^2} + \frac{155 - 786z}{1596} + \frac{33z - 22}{19(1-z)^2} + \frac{552763}{2184924} \frac{1}{(1-z)^2}$$

De là, on tire:

$$E_n = \frac{90}{259} (n+1)H_{n+1} + \frac{431255}{2184924} n - \frac{3445733}{2184924}$$

Soit 
$$E_n = \frac{90}{259} (n+1) \ln(n+1) + 0.398n$$

ANNEXE VIII: Calcul théorique du nombre d'appels pour la médiane de 5

La récurrence du nombre d'appels est:

$$A_n = 1 + \sum_{k=1}^n \frac{(n-k)(n-k-1)(k-1)(k-2)}{C_n^5} (A_{k-1} + A_{n-k})$$

avec

$$\begin{cases} A_0 = 1 \\ A_1 = 1 \\ A_2 = 3 \\ A_3 = \frac{13}{3} \\ A_4 = \frac{17}{3} \end{cases}$$

En posant

$$A(z) = \sum_{n=0}^{+\infty} A_n z^n$$

nous obtenons:

$$A^{(5)}(z) = \frac{120}{(1-z)^6} + 120 \frac{A'''(z)}{(1-z)^2}$$

Posons maintenant:  $U(z) = A''(z)$

L'équation d'Euler devient:  $(\Psi - 4)(\Psi - X_1)(\Psi - X_2)U(z) = \frac{720}{(1-z)^4} + \frac{720}{(1-z)^3}$

avec  $X_1 = -\frac{7}{2} - \frac{1}{2}i\sqrt{71}$  et  $X_2 = -\frac{7}{2} + \frac{1}{2}i\sqrt{71}$

$$\begin{cases} (\Psi - X_1)S(z) = \frac{120}{(1-z)^3} \\ (\Psi - X_2)T(z) = S(z) \\ (\Psi - 4)U(z) = T(z) \end{cases}$$

avec

$$\begin{cases} S(0) = 13 - i\sqrt{71} \\ T(0) = 2 \\ U(0) = 6 \end{cases}$$

En résolvant les équations différentielles, on obtient:

$$S(z) = -\frac{i\sqrt{71}-13}{(1-z)^3}$$

$$T(z) = \frac{2}{(1-z)^3}$$

$$U(z) = \frac{2z}{(1-z)^4} + \frac{6}{(1-z)^4}$$

Par conséquent, avec  $A(0) = 1$  et  $A'(0) = 1$ :

$$A(z) = \frac{z}{(1-z)^2} + \frac{1}{3} \frac{1}{(1-z)^2}$$

Ainsi

$$A_n = \frac{4}{3}n + \frac{1}{3}$$

ANNEXE IX: Calcul théorique du nombre de partitions pour la médiane de 5

La récurrence du nombre de partitions est :

$$P_n = 1 + \sum_{k=1}^n \frac{(n-k)(n-k-1)(k-1)(k-2)}{C_n^5} (P_{k-1} + P_{n-k})$$

avec

$$\begin{cases} P_0 = 0 \\ P_1 = 0 \\ P_2 = 1 \\ P_3 = \frac{5}{3} \\ P_4 = \frac{7}{3} \end{cases}$$

On obtient, comme pour le nombre d'appels, l'équation différentielle :

$$P^{(5)}(z) = \frac{120}{(1-z)^6} + 120 \frac{P'''(z)}{(1-z)^2}$$

La résolution est identique à celle du calcul du nombre d'appels, seules les conditions initiales changent:

$$\text{avec } \begin{cases} S(0) = 13 - i\sqrt{71} \\ T(0) = 2 \\ U(0) = 2 \end{cases}$$

Les équations différentielles obtenues sont alors:

$$S(z) = -\frac{i\sqrt{71}-13}{(1-z)^3}$$

$$T(z) = \frac{2}{(1-z)^3}$$

$$U(z) = \frac{2z}{(1-z)^4} + \frac{2}{(1-z)^4}$$

Par conséquent, avec  $P(0) = 0$  et  $P'(0) = 0$ :

$$A(z) = \frac{z}{(1-z)^2} - \frac{1}{3} \frac{1}{(1-z)^2}$$

Ainsi

$$\boxed{P_n = \frac{2}{3}n - \frac{1}{3}}$$

ANNEXE X: Nombre de comparaisons du Quicksort Médiane de  $2t + 1$

L'équation (b) devient:

$$n(n-1)\dots(n-2t)C_n = (n+1)n\dots(n-2t) + 2 \frac{(2t+1)!}{t!^2} \sum_{k=1}^n \frac{(n-k)!(k-1)!}{(n-k-t)!(k-1-t)!} C_{k-1}$$

Traduisons en équation différentielle, en multipliant les deux membres par  $z^{n-2t}$  et en sommant sur  $n$ :

$$(1-z)^{2t+1} C^{(2t+1)}(z) = \frac{2(2t+1)!}{t!} (1-z)^t C^{(t)}(z) + \frac{(2t+2)!}{(1-z)^2} \quad (1)$$

C'est une équation d'Euler. Récrivons là en utilisant l'opérateur  $\Psi \equiv (1-z) \frac{d}{dz}$ , sachant que  $(1-z)^t C^{(t)}(z) = \Psi(\Psi+1)\dots(\Psi+t-1)C(z)$  :

$$\begin{aligned} (\Psi+2t)\dots\Psi C(z) &= \frac{2(2t+1)!}{t!} (\Psi+t-1)\dots\Psi C(z) + \frac{(2t+2)!}{(1-z)^2} \\ \Psi(\Psi+1)\dots(\Psi+t-1) \left[ (\Psi+t)\dots(\Psi+2t) - \frac{(2t+2)!}{(t+1)!} \right] C(z) &= \frac{(2t+2)!}{(1-z)^2} \end{aligned}$$

Les racines du polynôme en  $\Psi$  formé par le membre de gauche sont  $0, -1 \dots (1-t)$  ainsi que les racines de  $P(\Psi) = (\Psi+t)\dots(\Psi+2t) - \frac{(2t+2)!}{(t+1)!}$ . Nous pouvons remarquer que la racine de plus grande partie réelle de  $P$  est 2. Le terme principal de la solution homogène sera donc en  $(1-z)^{-2}$ . Cela donnera un terme dominant pour la solution générale en  $\frac{1}{(1-z)^2} \ln\left(\frac{1}{1-z}\right)$ . Nous allons donc chercher une solution particulière  $f(z) = \alpha \frac{1}{(1-z)^2} \ln\left(\frac{1}{1-z}\right)$ .

On montre par récurrence que:

$$f^{(k)}(z) = \frac{\alpha(k+1)!}{(1-z)^{k+2}} \ln\left(\frac{1}{1-z}\right) + \frac{\alpha(k+1)!}{(1-z)^{k+2}} (H_{k+1} - 1)$$

On réinjecte  $f^{(k)}(z)$  dans l'équation (1) et on obtient:

$$\alpha = \frac{1}{H_{2t+2} - H_{t+1}}$$

Le terme dominant de la solution de l'équation avec conditions initiales sera  $f(z)$  et les termes suivants, qui vont eux dépendre effectivement des conditions initiales seront de la forme  $(1-z)^a$  avec  $a \geq -2$ .

Mais pour un  $n$  assez grand, les termes de la forme  $(1-z)^a$  avec  $a > -2$  deviennent négligeables lors du passage de  $C(z)$  à  $C_n$ . Une bonne approximation de  $C(z)$  sera donc:

$$C(z) = \alpha \frac{1}{(1-z)^2} \ln\left(\frac{1}{1-z}\right) + K_0 \frac{1}{(1-z)^2}$$

Il suffit maintenant d'extraire  $C_n$ :

$$C_n = \alpha(n+1)(H_{n+1}) + K_0(n+1) \quad \text{avec} \quad \alpha = \frac{1}{H_{2t+2} - H_{t+1}}$$

Les deux extrémums de la fonction  $\alpha(t) = \frac{1}{H_{2t+2} - H_{t+1}}$  sont :

- pour  $t = 0$ ,  $\alpha(t) = 2$
- pour  $t = \frac{n-1}{2}$ ,  $\alpha(t) = \frac{1}{\ln 2}$

A présent, résolvons (c) de la même manière que (b). L'équation différentielle obtenue est:

$$(1-z)^{2t+1} C^{(2t+1)}(z) = \frac{(2t+2)!}{(t+1)!} (1-z)^t C^{(t)}(z) + \lambda \frac{(2t+1)!}{(1-z)} \quad (2)$$

Cette fois le terme dominant de la solution est en  $\frac{z}{(1-z)^2}$ .

Nous allons donc calculer une solution particulière  $g(z) = \beta \frac{z}{(1-z)^2}$ .

On montre par récurrence que:

$$g^{(k)}(z) = \beta \left[ k \frac{k!}{(1-z)^{k+1}} + \frac{z(k+1)!}{(1-z)^{k+2}} \right]$$

En réinjectant dans (2), on trouve:

$$\beta = \lambda$$

Ainsi, nous obtenons la solution:

$$C_n = \lambda n + K_0(n+1)$$

Nous pouvons aussi déduire de ce résultat, le nombre d'appels  $A_n$  et de partitions  $P_n$  effectués par le Quicksort Médiane de  $2t+1$  éléments.

En prenant  $\lambda = 1$ , on obtient alors:

$$\begin{cases} A_n = n + A_0(n+1) \\ P_n = n + P_0(n+1) \end{cases}$$

Nous allons maintenant donner une approximation de  $E_n$  en fonction des résultats obtenus ci-dessus. Soit  $EP_n$  le nombre d'échanges effectués durant les partitions:

$$EP_n = 2 + \sum_{k=1}^n \frac{C_{k-1}^t C_{n-k}^t (n-k)(k-1)}{C_n^{2t+1} (n-1)}$$

En dégagant le terme dominant de cette expression, c'est-à-dire en extrayant le terme linéaire, nous pourrions calculer le coefficient du terme dominant du nombre d'échanges effectués par le Quicksort médiane de  $2t+1$  éléments.

Tout d'abord, en prenant  $C_{k-1}^t \sim \frac{(k-1)^t}{t!}$  et  $C_{n-k}^t \sim \frac{(n-k)^t}{t!}$ , on obtient

$$EP_n \sim \frac{1}{(n-1)t!^2 C_n^{2t+1}} \sum_{k=1}^n (k-1)^t (n-k)^t (k-1)(n-k)$$

Posons

$$EP'_n = \sum_{k=1}^n (k-1)^{t+1} (n-k)^{t+1}$$

$$EP''_n = \sum_{k=1}^n ((k-1)(n-k))^{t+1}$$

$$EP'_n \sim \sum_{k=1}^n (nk - k^2)^{t+1}$$

La formule du binôme de Newton nous donne

$$EP'_n \sim \sum_{k=1}^n \sum_{i=0}^{t+1} (-1)^i C_{t+1}^i (nk)^{t+1-i} (k^2)^i$$

$$\begin{aligned} EP'_n &\sim C_{t+1}^0 \sum_{k=1}^n (nk)^{t+1} - C_{t+1}^1 \sum_{k=1}^n (nk)^t k^2 + C_{t+1}^2 \sum_{k=1}^n (nk)^{t-1} k^4 \\ &\quad - \dots + (-1)^t C_{t+1}^t \sum_{k=1}^n (nk) k^{2t} + (-1)^{t+1} C_{t+1}^{t+1} \sum_{k=1}^n k^{2t+2} \end{aligned}$$

Or  $\sum_{k=1}^n k^t \sim \frac{n^{t+1}}{t+1}$  donc

$$\begin{aligned} EP'_n &\sim C_{t+1}^0 n^{t+1} \left( \frac{n^{t+2}}{t+2} \right) - C_{t+1}^1 \left( \frac{n^{t+3}}{t+3} \right) + C_{t+1}^2 \left( \frac{n^{t+4}}{t+4} \right) \\ &\quad - \dots + (-1)^t C_{t+1}^t \left( \frac{n^{2t+2}}{2t+2} \right) + (-1)^{t+1} C_{t+1}^{t+1} \left( \frac{n^{2t+3}}{2t+3} \right) \end{aligned}$$

$$EP'_n \sim \left( \sum_{i=0}^{t+1} (-1)^i \frac{C_{t+1}^i}{t+2+i} \right) n^{2t+3}$$

Par conséquent,

$$EP_n \sim \frac{n^{2t+2}}{(n-1)t! C_n^{2t+1}} \sum_{i=0}^{t+1} (-1)^i \frac{C_{t+1}^i}{t+2+i}$$

Et  $C_n^{2t+1} \sim \frac{n^{2t+2}}{(2t+1)!}$  donc

$$EP_n \sim \left( \frac{(2t+1)!}{t!^2} \sum_{i=0}^{t+1} (-1)^i \frac{C_{t+1}^i}{t+2+i} \right) n$$

Enfin,

$$EP_n \sim \frac{(2t+1)!}{t!^2 (t+2) C_{2t+3}^{t+2}} n$$

$$EP_n \sim \frac{t+1}{2(2t+3)} n$$

Par conséquent,  $E_n$  est bien une combinaison linéaire de  $C_n$  et

$$\alpha_e = \frac{t+1}{2(2t+3)} \alpha$$

**ANNEXE XI: Calculs théoriques relatifs au tri par insertion**

Cette étude est effectuée dans le cas moyen en prenant comme hypothèse la distinction des clés à trier.

a) Le nombre de comparaisons:

$$C_n = \sum_{i=2}^n \left(1 + \frac{i-1}{2}\right)$$

$$C_n = \frac{n(n+3)}{4} - 1$$

avec un écart type  $Ec_n = \sqrt{\frac{(n-2)(2n^2+7n+9)}{72}} = O\left(\frac{n\sqrt{n}}{6}\right)$ .

b) Le nombre de transferts (affectations) :

Celui-ci est donné par le nombre d'inversions:  $I_n = \frac{n(n-1)}{4}$   
auquel il faut ajouter les  $2(n-1)$  affectations correspondant à  $v = a[i]$  et  $a[j] = v$   
cela donne donc  $T_n = I_n + 2(n-1)$ .

Soit

$$T_n = \frac{n(n+7)}{4} - 2$$

L'écart type reste bien sûr en  $O\left(\frac{n\sqrt{n}}{6}\right)$

Lors des calculs du nombre d'échanges pour Quicksort, étant donné la difficulté d'évaluer le temps d'exécution précis des différents paramètres du programme, nous assimilerons un échange dans le Quicksort à trois transferts dans le tri par insertion.

Le nombre d'échanges du tri par insertion sera donc:

$$E_n = \frac{n(n+7) - 8}{12}$$

**ANNEXE XII: Calculs théoriques relatifs au tri par insertion dichotomique**

Cette étude est effectuée dans le pire des cas en prenant comme hypothèse la distinction des clés à trier.

a) Le nombre de comparaisons :

Calculons tout d'abord le nombre de comparaisons dues à l'insertion dichotomique d'un élément dans le pire des cas vérifiant la récurrence :

$$B_n = B_{\lceil \frac{n}{2} \rceil} + 1 \quad \text{avec} \quad B_1 = 0$$

Considérons alors  $2^{N-1} < n \leq 2^N$ :

$$B_n = \sum_{i=1}^N 1 = N$$

$$B_n = \lceil \log_2 n \rceil$$

Par conséquent, le nombre de comparaisons pour trier devient:

$$C_n = \sum_{k=2}^n \lceil \log_2 k \rceil$$

En posant  $2^{N-1} < n \leq 2^N$ , nous pouvons vérifier que:

$$C_n = N(n - 2^{N-1}) + (N-1)(2^{N-1} - 2^{N-2}) + \dots + 1(2^1 - 2^0)$$

$$C_n = N(n - 2^{N-1}) + \sum_{i=1}^{N-1} i2^{i-1}$$

or

$$\sum_{i=1}^{N-1} i2^{i-1} = (N-2)2^{N-1} + 1$$

donc

$$C_n = nN - 2^N + 1$$

Par définition, nous avons  $N = \lceil \log_2 n \rceil$ , donc

$$C_n = n \lceil \log_2 n \rceil - 2^{\lceil \log_2 n \rceil} + 1$$

Posons  $\lceil \log_2 n \rceil = \log_2 n + \theta$  avec  $\theta \in [0, 1[$ , l'équation devient:

$$C_n = n \log_2 n + n(\theta - 2^\theta) + 1$$

Le maximum de la fonction monotone  $f(\theta) = \theta - 2^\theta$  s'obtient en résolvant  $f'(\theta) = 0$ , soit:

$$f_{max}(\theta_{max}) = -\frac{\ln \ln 2 + 1}{\ln 2} \quad \text{pour} \quad \theta_{max} = -\frac{\ln \ln 2}{\ln 2}$$

Par conséquent, dans le pire des cas, nous pouvons affirmer que:

$$n \log_2 n - n + 1 \leq C_n \leq n \log_2 n - 0,914n + 1$$

Le nombre d'échanges est évidemment le même que celui du tri par insertion.

Cependant, il faut préciser que lorsque des clés sont égales, l'algorithme effectue plus de comparaisons, ce qui peut être gênant dans son utilisation comme tri de coupure avec Quicksort.

I/ Pour les comparaisons :

De l'équation (1), Annexe II page 21, on tire la récurrence:

$$(n+1)C_{n+1} = 2 \sum_{k=0}^n C_k + \sum_{k=0}^n T_k$$

$$(n+1)C_{n+1} = (n+2)C_n + T_n$$

De la même manière on tire de l'équation (2), Annexe II page 21:

$$T_{n-1} = \frac{12}{7}n + \frac{8}{3}$$

On obtient donc:

$$C_n = \frac{n+1}{n}C_{n-1} + \frac{12}{7}n + \frac{8}{3}$$

Résolvons cette récurrence en itérant jusqu'à m et en sommant:

$$\frac{C_n}{n+1} = \frac{C_{m-1}}{m} + \sum_{i=m+1}^{n+1} \left( \frac{12}{7} \frac{1}{i} + \frac{8}{3} \frac{1}{i(i-1)} \right)$$

$$\text{où } C_{m-1} = \frac{(m-1)(m+2)-4}{4}$$

$$\frac{C_n}{n+1} = \frac{12}{7}(H_{n+1} - H_m) + \frac{8}{3} \left( \frac{1}{m} - \frac{1}{n+1} \right) + \frac{(m-1)(m+2)-4}{4m}$$

Par conséquent:

$$C_n = \frac{12}{7}(n+1)H_{n+1} + (n+1)f(m)$$

$$\text{où } f(m) = \frac{3m^2 + 3m + 14}{12m} - \frac{12}{7}H_m$$

II/ Pour les échanges :

En suivant le même procédé que pour le calcul du nombre de comparaisons, on obtient les résultats suivants:

$$\frac{E_n}{n+1} = \frac{E_{n-1}}{n} + \frac{12}{35} \frac{1}{(n+1)} + \frac{24}{15} \frac{1}{(n+1)n}$$

En itérant jusqu'à  $m$ , en sommant et en prenant  $E_{m-1} = \frac{(m+6)(m-1)-8}{12}$ :

$$E_n = \frac{12}{35}(n+1)H_{n+1} + (n+1)f(m)$$

avec  $f(m) = \frac{5m^2 + 25m + 56}{60m} - \frac{12}{35}H_m$

III/ Pour les appels et les partitions :

$$\frac{A_n}{n+1} = \frac{A_{n-1}}{n} + \frac{1}{(n+1)n}$$

En itérant jusqu'à  $m$  et en sommant:

$$\frac{A_n}{n+1} = \frac{A_{m-1}}{m} + \sum_{i=m}^n \frac{1}{i(i+1)}$$

$$\frac{A_n}{n+1} = \frac{2}{m} - \frac{1}{n+1}$$

d'où, puisque  $A_{m-1} = 1$ ,

$$A_n = \frac{2(n+1)}{m} - 1$$

Dans ce cas, le nombre de partitions est donc, avec  $P_{m-1} = 0$ :  $P_n = A_n = \frac{n+1}{m} - 1$

Nous avons trouvé en pratique un  $m$  optimum égal à 7.

ANNEXE XIV: *Calculs théoriques du point de coupure pour Médiane de 5*

I/ Pour les comparaisons :

En procédant de la même manière que pour le calcul du point de coupure pour la médiane de 3, on obtient le nombre de comparaisons suivant:

$$\frac{C_n}{n+1} = \frac{C_{n-1}}{n} + \frac{60}{37} \frac{1}{n+1} + \frac{6}{(n+1)n}$$

En itérant jusqu'à  $m$  et en sommant, avec  $C_{m-1} = \frac{(m-1)(m+2)-4}{4}$ , on obtient:

$$C_n = \frac{60}{37}(n+1)H_{n+1} + (n+1)f(m)$$

$$\text{où } f(m) = \frac{m^2 + m + 18}{4m} - \frac{60}{37}H_m$$

II/ Pour les échanges :

Le nombre d'échanges lui est donné par la récurrence suivante:

$$\begin{aligned} \frac{E_n}{n+1} = \frac{E_{n-1}}{n} - \frac{1}{29526} \left[ -222 \frac{(n-3)(n-4)(n-5)}{(n-2)(n-1)n(n+1)} + 1369 \frac{(n-3)(n-4)}{(n-1)n(n+1)} - 18241 \frac{n-3}{n(n+1)} \right. \\ \left. + 27354 \frac{n-1}{n(n+1)} \right] \end{aligned}$$

En itérant jusqu'à  $m$  et en sommant, avec  $E_{m-1} = \frac{(m-1)(m+6)-8}{12}$ , on obtient:

$$E_n = \frac{90}{259}(n+1)H_{n+1} + (n+1)f(m) - \frac{7051}{9842} + O\left(\frac{1}{n}\right)$$

$$\text{où } f(m) = \frac{(m-1)(m+6)-8}{12} - \frac{90}{259}H_m + \frac{7051}{9842} \frac{1}{m} + \frac{1}{14} \frac{1}{m(m-1)} + \frac{2}{133} \frac{1}{m(m-1)(m-2)}$$

III/ Pour les appels et les partitions :

On a la récurrence:

$$\frac{A_n}{n+1} = \frac{A_{n-1}}{n} + \frac{1}{n(n+1)}$$

On obtient donc, par itération jusqu'à  $m$  et sommation, avec  $A_{m-1} = 1$ :

$$A_n = \frac{2}{m}(n+1) - 1$$

et pour les partitions, avec  $P_{m-1} = 0$ :

$$P_n = \frac{n+1}{m} - 1$$

Nous avons trouvé expérimentalement un  $m$  optimum égal à 9.

Ce problème a été suggéré à P. Flajolet par C. Martinez en 1997.

Reprenons l'équation (I) (page 5):

$$\frac{C_n}{n+1} = \alpha(H_{n+1} - 1) + \frac{\lambda n}{n+1} + K_0$$

$$\frac{C_{n-1}}{n} = \alpha(H_n - 1) + \frac{\lambda(n-1)}{n} + K_0$$

Par soustraction, on obtient:

$$\boxed{\frac{C_n}{n+1} = \frac{C_{n-1}}{n} + \frac{\alpha}{n+1} + \frac{\lambda}{n(n+1)}}$$

En itérant jusqu'à  $m$  et en sommant: ( $m \in [0, n]$ , et  $m \geq 2t + 1$ )  
Soit  $C_n^*$  le nombre de comparaisons effectuées par notre tri utilisé à la coupure.

$$\frac{C_n}{n+1} = \frac{C_m^*}{m+1} + \sum_{k=m+2}^{n+1} \frac{\alpha}{k} + \sum_{k=m+1}^n \frac{\lambda}{k(k+1)}$$

$$\frac{C_n}{n+1} = \alpha(H_{n+1} - H_{m+1}) + \frac{\lambda}{m+1} - \frac{\lambda}{n+1} + \frac{C_m^*}{m+1}$$

$$\boxed{C_n = \alpha(n+1)H_{n+1} + f(m)(n+1) - \lambda} \quad (a)$$

avec  $\boxed{f(m) = \frac{C_m^*}{m+1} + \frac{\lambda}{m+1} - \alpha H_{m+1}}$  (b)

On remarque que:

- pour  $m = 0$ , on retombe sur notre équation (I) (page 5):

$$\boxed{C_n = \alpha(n+1)(H_{n+1} - 1) + \lambda n + C_0(n+1) \quad \text{où} \quad K_0 = C_0}$$

Les conditions initiales, comme nous l'avons expliqué, influent donc précisément sur le terme en  $(n+1)$ .

- pour  $m = n$ :  $C_n = C_n^*$ .

Pour calculer l'optimum de la fonction  $f$ , il suffit de résoudre  $f'(m) = 0$ .  
 $\alpha$  et  $\beta$  dépendent de  $t$  uniquement donc:

$$f'(m) = \frac{C_m^{*'}(m+1) - C_m^* - \lambda}{(m+1)^2} - \frac{\alpha}{m+1}$$

$$\boxed{f'(m) = \frac{(m+1)(C_m^{*'} - \alpha) - C_m^* - \lambda}{(m+1)^2}}$$

Prenons maintenant:  $f'(m) = 0$

$$(m+1)(C_m^{*'} - \alpha) - C_m^* - \lambda = 0 \quad (c)$$

Pour le tri par insertion:  $C_m^* = \frac{m(m+3)-4}{4}$  donc  $C_m^{*'} = \frac{2m+3}{4}$

Réinjectons dans (c) pour obtenir:  $m^2 + m(2 - 4\alpha) + (7 - 4\alpha - 4\lambda) = 0$

$$\Delta = (2 - 4\alpha)^2 - 4(7 - 4\alpha - 4\lambda)$$

Nous avons deux solutions dont une seule est positive:

$$m_{opt} = \frac{-(2 - 4\alpha) + \sqrt{\Delta}}{2}$$

Résumons: Le nombre de comparaisons du Quicksort médiane de  $2t + 1$  éléments avec coupure à  $m_{opt}$  par le tri par insertion est:

$$C_n = \alpha(n+1)(H_{n+1}) + f(m_{opt})(n+1) - \lambda$$

$$\text{avec } \alpha = \frac{1}{H_{2t+2} - H_{t+1}}$$

$$f(m_{opt}) = \frac{C_{m_{opt}}^*}{m+1} + \frac{\lambda}{m+1} - \alpha H_{m+1}$$

$$\text{avec } C_{m_{opt}}^* = \frac{m_{opt}(m_{opt}+3)-4}{4}$$

$\lambda$  le nombre de comparaisons nécessaires pour la recherche du médian des  $2t + 1$  éléments, soit ici,  $\frac{3}{2}(2t + 1)$

$$m_{opt} = -1 + 2\alpha + \frac{1}{2}\sqrt{16\alpha^2 + 8\alpha - 24 + 16\lambda}$$

Mais l'on peut bien sûr effectuer ces calculs pour un  $C_m^*$  différent tel que le tri par insertion dichotomique, le tri optimal ... Et de même,  $\lambda$  peut être modifié pour obtenir des valeurs plus précises (tant qu'il reste constant par rapport à  $n$ ).

Nous pouvons effectuer le même raisonnement pour calculer  $A_n$  et  $P_n$ .

$$\text{Nous trouvons: } \begin{cases} A_n = \frac{2}{m+1}(n+1) - 1 \\ P_n = \frac{1}{m+1}(n+1) - 1 \end{cases} \text{ car } A_m^* = 1 \text{ et } P_m^* = 0.$$

**ANNEXE XVI: Quicksort à trois paliers**

Nous allons donc calculer les deux points de coupure entre les trois méthodes:

a) Le nombre de comparaisons :

Partons donc de la récurrence suivante:

$$\frac{C_n}{n+1} = \frac{C_{n-1}}{n} + \frac{60}{37} \frac{1}{n+1} + \frac{6}{n(n+1)}$$

On itère jusqu'à  $m_1$  puis l'équation devient:

$$\frac{C_{m_1}}{m_1+1} = \frac{C_{m_1-1}}{m_1} + \frac{12}{7} \frac{1}{m_1+1} + \frac{8}{3} \frac{1}{m_1(m_1+1)}$$

De même, on itère jusqu'à  $m_2$  où:

$$C_{m_2} = \frac{m_2(m_2-3)-4}{4}$$

En sommant le tout on obtient donc:

$$C_n = \frac{60}{37}(n+1)H_{n+1} + f(m_1)(n+1) + f(m_2)(n+1) - 6$$

avec:

$$f(m_1) = \frac{24}{259}H_{m_1+1} + \frac{10}{3} \frac{1}{m_1+1}$$

$$f(m_2) = -\frac{12}{7}H_{m_2+1} + \frac{3m_2^2+3m_2+14}{12m_2}$$

b) Le nombre d'échanges :

En suivant le même raisonnement que précédemment, les trois récurrences deviennent:

$$\frac{E_n}{n+1} = \frac{E_{n-1}}{n} - \frac{1}{29526} \left[ -222 \frac{(n-3)(n-4)(n-5)}{(n-2)(n-1)n(n+1)} + 1369 \frac{(n-3)(n-4)}{(n-1)n(n+1)} - 18241 \frac{n-3}{n(n+1)} \right]$$

$$\frac{E_{m_1}}{m_1+1} = \frac{E_{m_1-1}}{m_1} + \frac{12}{35} \frac{1}{m_1+1} + \frac{24}{15} \frac{1}{(m_1+1)m_1}$$

$$E_{m_2} = \frac{m_2(m_2+7)-8}{12}$$

Par conséquent,

$$E_n = \frac{90}{259}(n+1)H_{n+1} + (n+1)f(m_1) + (n+1)f(m_2) + O(1)$$

avec:

$$f(m_1) = -\frac{56311999}{88570} \frac{1}{m_1+1} - \frac{6}{1295} H_{m_1+1}$$

$$f(m_2) = \frac{5m_2^2+25m_2+56}{60} \frac{1}{m_2} - \frac{12}{35} H_{m_2+1}$$

c) Le nombre d'appels et de partitions :

En suivant le même raisonnement que précédemment, les trois récurrences deviennent:

$$\frac{A_n}{n+1} = \frac{A_{n-1}}{n} + \frac{1}{n(n+1)}$$

$$\frac{A_{m_1}}{m_1+1} = \frac{A_{m_1-1}}{m_1} + \frac{1}{m_1(m_1+1)}$$

$$A_{m_2} = 1$$

Par conséquent,  $A_n = f(m_2)(n+1) - 1$  avec  $f(m_2) = \frac{2}{m_2+1}$

Le nombre de partitions s'obtient juste en changeant  $A_{m_2} = 1$  par  $P_{m_2} = 0$ .

Cela nous donne:  $P_n = f(m_2)(n+1) - 1$  avec  $f(m_2) = \frac{1}{m_2+1}$

Nous avons trouvé expérimentalement les points de coupure optimaux  $m_1 \approx 38$  et  $m_2 \approx 9$ .

ANNEXE XVII: Médiane de 9 approchée + Résultats

La médiane de 9 peut être approchée en prenant la médiane des médianes de 3 éléments. La médiane M de 9 approchée donne trois types de solutions:

- M > 3 éléments et M < 5 éléments avec une probabilité de  $\frac{1}{4}$
- M > 5 éléments et M < 3 éléments avec une probabilité de  $\frac{1}{4}$
- M > 4 éléments et M < 4 éléments (médiane exacte) avec probabilité de  $\frac{1}{2}$

L'équation de Quicksort sera alors:

$$C_n = n + 1 + \frac{32}{3} + \sum_{k=1}^n \left( \frac{C_{k-1}^3 C_{n-k}^5 + C_{k-1}^5 C_{n-k}^3 + 2C_{k-1}^4 C_{n-k}^4}{4C_n^9} \right) (C_{k-1} + C_{n-k})$$

Ce qui nous donnerait:

En posant  $C(z) = \sum_{n=0}^{+\infty} C_n z^n$ :

$$C^{(9)}(z) = \frac{10!}{(1-z)^{11}} + \frac{32}{3} \frac{9!}{(1-z)^{10}} + \frac{9!}{2} \left( \frac{1}{6} \frac{C^{(3)}(z)}{(1-z)^6} + \frac{32}{3} \frac{C^{(4)}(z)}{(1-z)^5} + \frac{1}{120} \frac{C^{(5)}(z)}{(1-z)^4} \right)$$

Il suffit donc de résoudre cette équation différentielle et d'en retirer  $C_n$  pour obtenir le nombre de comparaisons du Quicksort médiane de 9 approchée. Les résultats de cette technique appliquée au Quicksort à 3 paliers ainsi que ceux des autres variantes donnent, pour  $n = 10^6$ , les temps d'exécution suivants (PC Dell, Processeur Intel Pentium III 450 MHz, Java JDK 1.2):

QS(0)	QS(1)	QS(2)	QS(3)	QS(3')
1368 ms	1292 ms	1289 ms	1284 ms	1283 ms

avec  $\left\{ \begin{array}{l} QS(0) : \text{version (0) du Quicksort} \\ QS(1) : \text{Quicksort médiane de 3 avec coupure à } m = 7 \\ QS(2) : \text{Quicksort médiane de 5 avec coupure à } m = 9 \\ QS(3) : \text{Quicksort à 3 paliers} \\ QS(3') : \text{Quicksort à 3 paliers avec la médiane de 9 approchée au palier 3} \end{array} \right.$

Et pour  $n = 16 * 10^6$ :

QS(0)	QS(3)
30 s	25 s

Nous voyons ici que le gain en temps devient considérable.

ANNEXE XVIII: Tableaux de résultats

Quicksort Médiane de 3

Taille n	Nombre de tirages	Nbre théorique de comparaisons	Nombre pratique de comparaisons
1000	1000	11600	11594
10000	100	155359	155540
100000	10	1948166	1968307

*Les comparaisons*

Taille n	Nombre de tirages	Nombre théorique d'échanges	Nombre pratique d'échanges
1000	1000	2853	2857
10000	100	36401	36416
100000	10	442933	442573

*Les échanges*

Taille n	Nombre de tirages	Nombre théorique d'appels	Nombre pratique d'appels
1000	1000	1143	1143
10000	100	11429	11429
100000	10	114285	114259

*Les appels*

Taille n	Nombre de tirages	Nombre théorique de partitions	Nombre pratique de partitions
1000	1000	571	571
10000	100	5714	5713
100000	10	57142	57128

*Les partitions*

Quicksort Médiane de 5

Taille n	Nombre de tirages	Nbre théorique de comparaisons	Nombre pratique de comparaisons
1000	1000	11864	11882
10000	100	155984	156295
100000	10	1933081	1939740

*Les comparaisons*

Taille n	Nombre de tirages	Nombre théorique d'échanges	Nombre pratique d'échanges
1000	1000	2801	2779
10000	100	35988	35777
100000	10	439867	437484

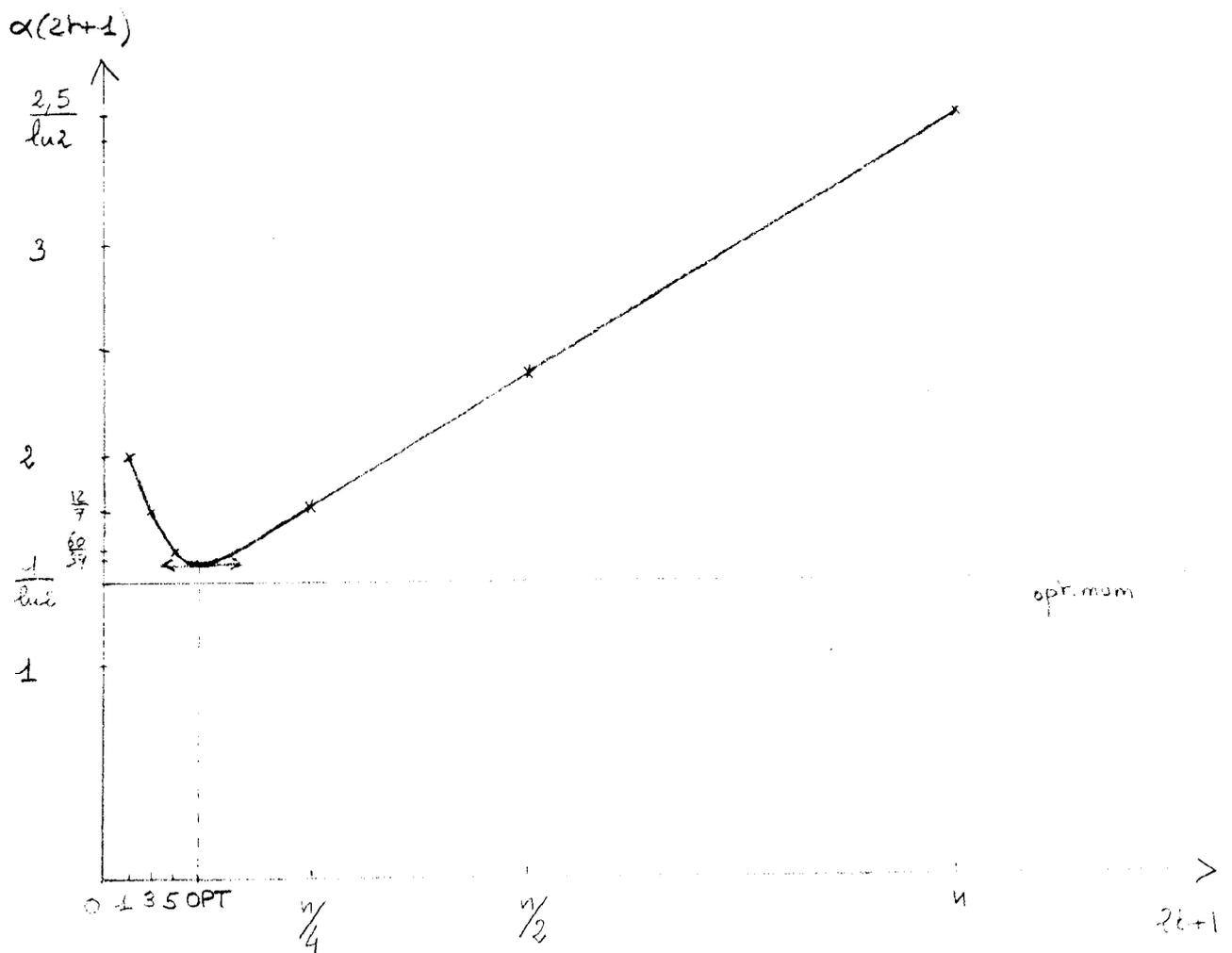
*Les échanges*

Taille n	Nombre de tirages	Nombre théorique d'appels	Nombre pratique d'appels
1000	1000	1334	1334
10000	100	13334	13334
100000	10	133334	133347

*Les appels*

Taille n	Nombre de tirages	Nombre théorique de partitions	Nombre pratique de partitions
1000	1000	667	666
10000	100	6667	6665
100000	10	66667	66668

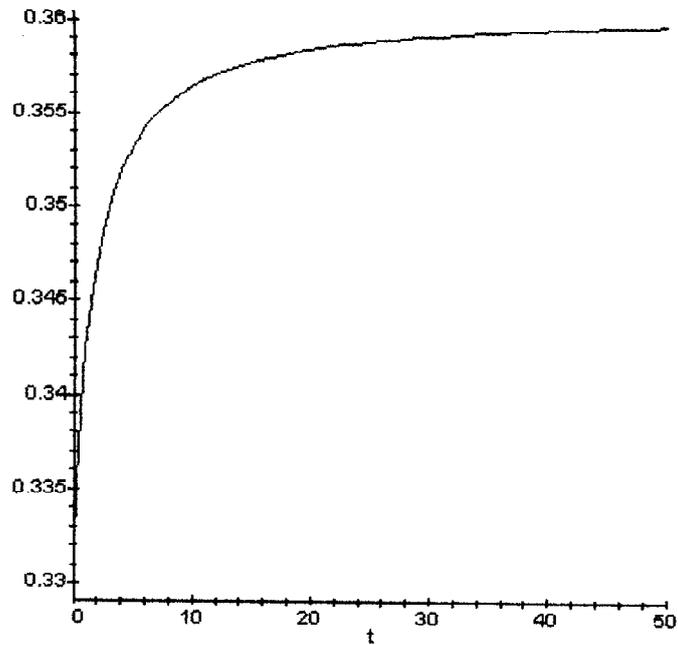
*Les partitions*



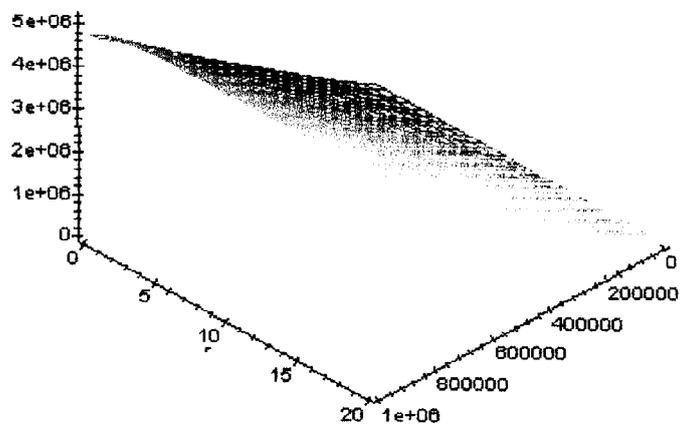
Graph de l'évolution de  $\alpha$  en fonction de  $2t+1$ .  
 On peut en déduire immédiatement  $C_u$  en fonction de  $2t+1$   
 en multipliant les valeurs de l'ordonnée par le terme  $(n+1)H_{n+1}$ .

Graph I

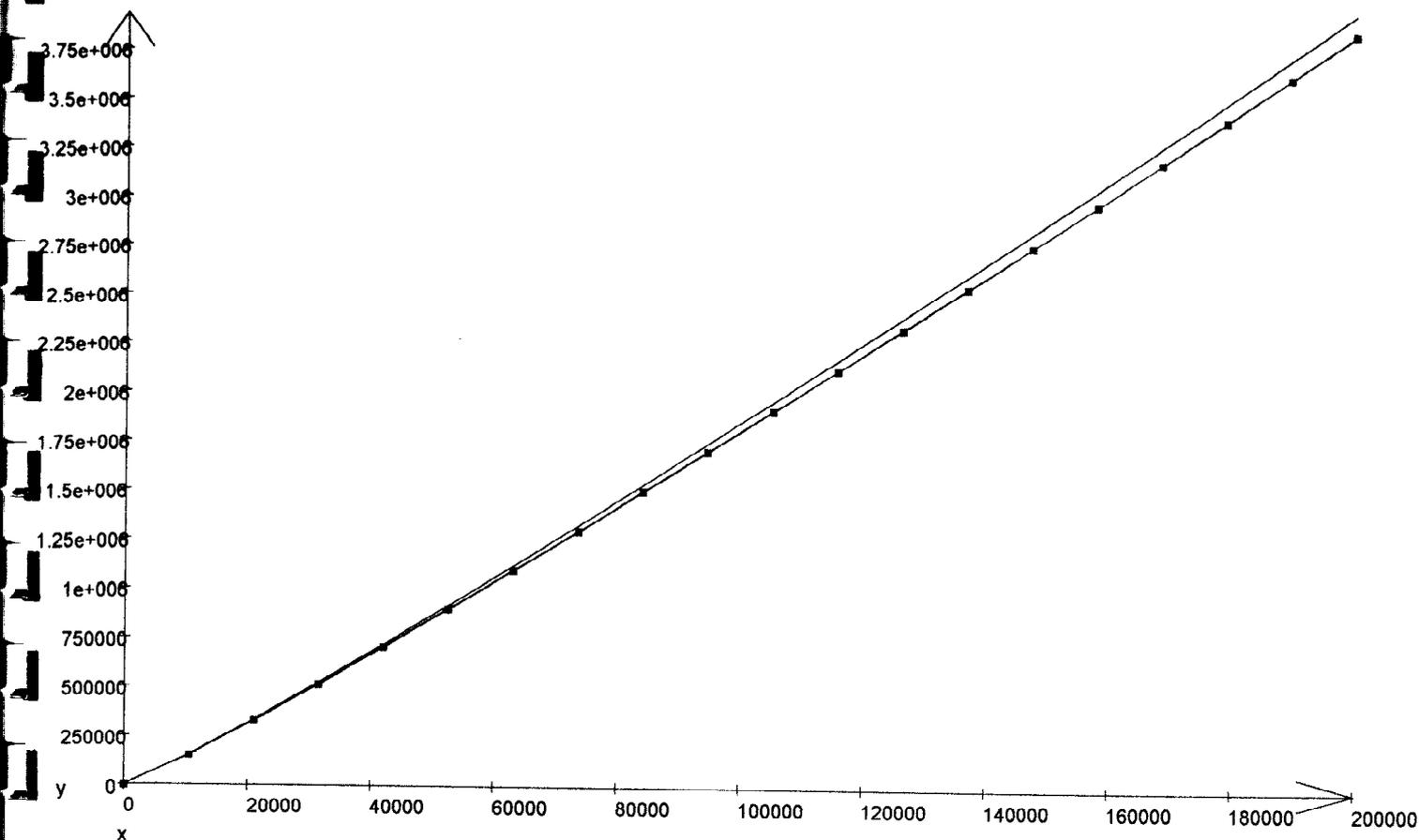
## GRAPHE II



Graph of  $\alpha_c$  (the coefficient of the dominant term of  $E_n$ ) in function of  $t$



Graph of  $E_n$ , the number of exchanges, in function of  $n$  and of  $t$  for the Quicksort médiane de  $2t+1$



En rouge : QuickSort médiane de 3 avec coupure

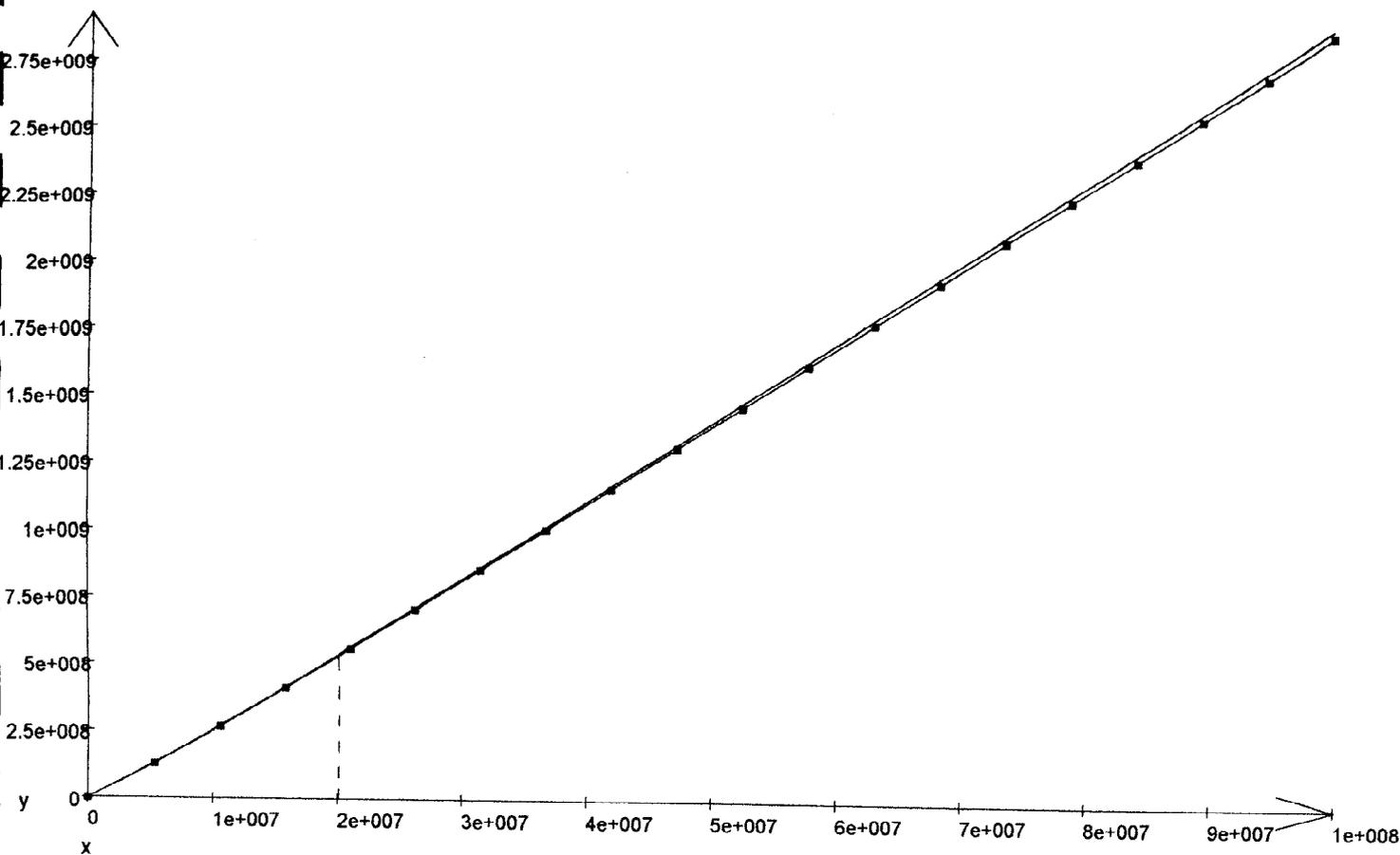
En vert et en bleu : QuickSort médiane de 5 avec coupure  
(pointillés) QuickSort médiane de 7 avec coupure

En abscisse :  $n$

En ordonnée :  $C_n$  le nombre de comparaisons

Ce graphique traduit aussi bien les résultats pratiques, que théoriques.

Graph III



En rouge : QuickSort médiane de 5 avec coupure

En vert pointillé : QuickSort médiane de 7 avec coupure

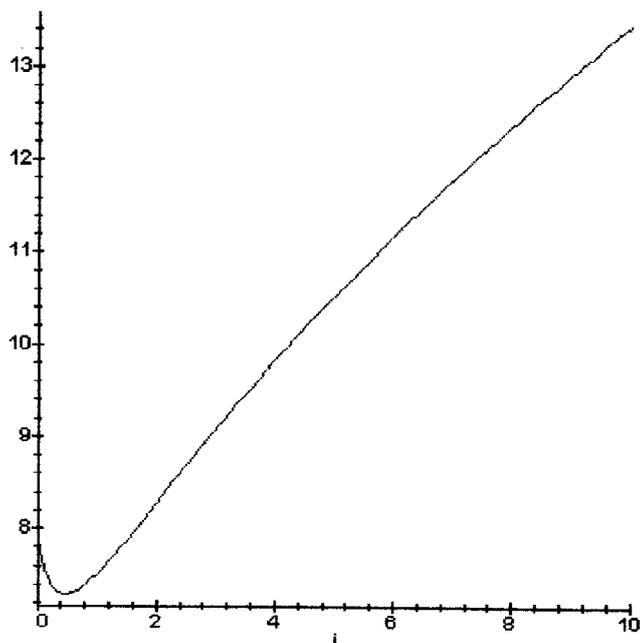
En abscisse :  $u$

En ordonnée :  $C_u$

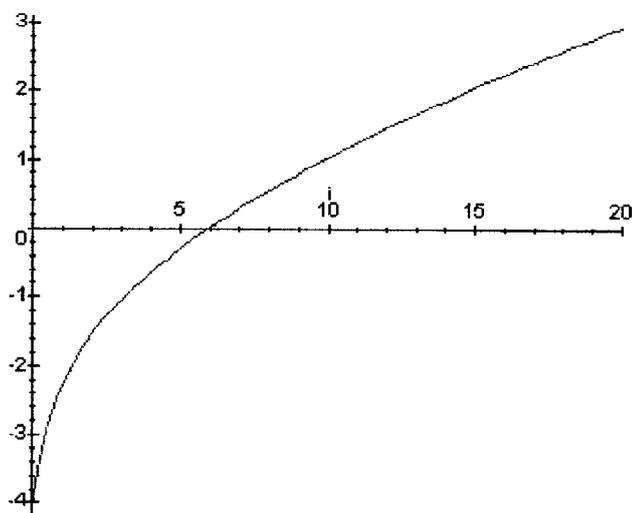
Graphes du nombre de comparaisons  $C_u$  pour le  
QuickSort médiane de 5 et le QuickSort médiane de 7

Graphes IV

## GRAPHE V

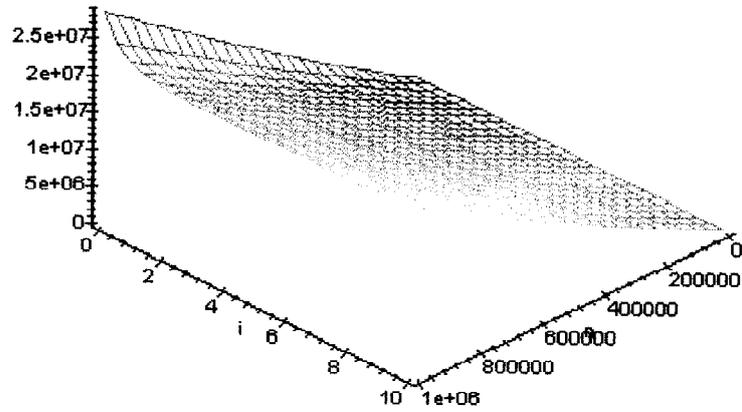


Graphe des points de coupure optimaux m en fonction de t pour le Quicksort médiane de  $2t+1$  éléments

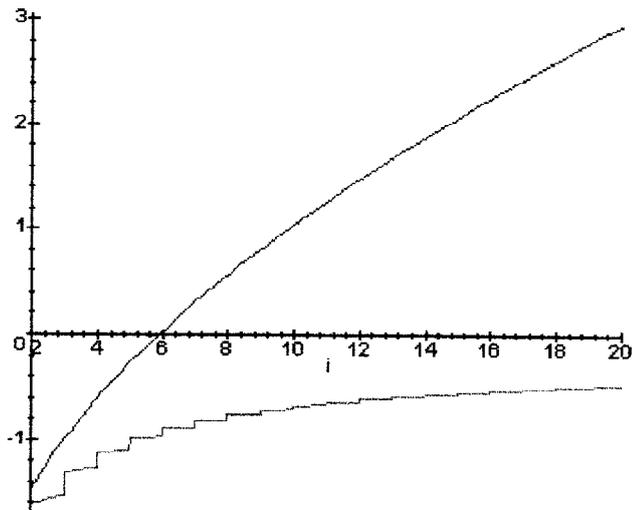


Graphe de  $f(m)$  avec m optimal en fonction de t pour le Quicksort médiane de  $2t+1$  avec point de coupure à m

## GRAPHE VI

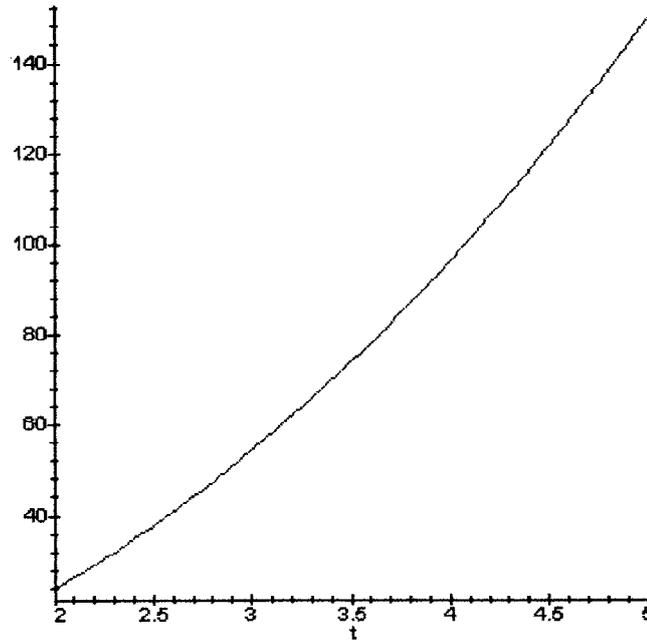


Graphe de  $C_n$ , le nombre de comparaisons, en fonction de  $n$  et de  $t$ , pour le Quicksort médiane de  $2t+1$  éléments avec point de coupure à  $m$

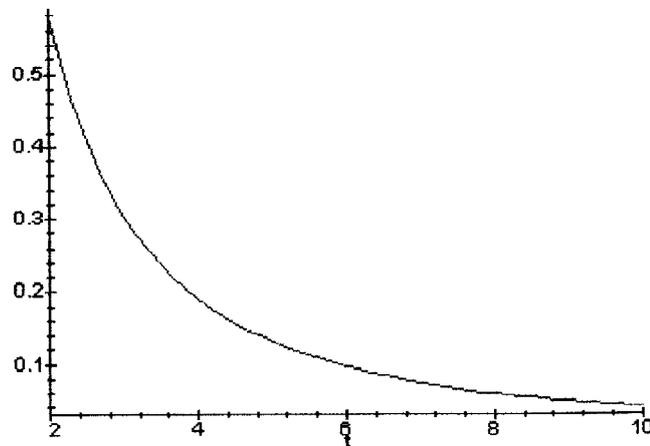


Graphe des coefficients des termes linéaires du Quicksort à  $t$  paliers (en vert) et du Quicksort médiane de  $2t+1$  éléments (en rouge)

## GRAPHE VII

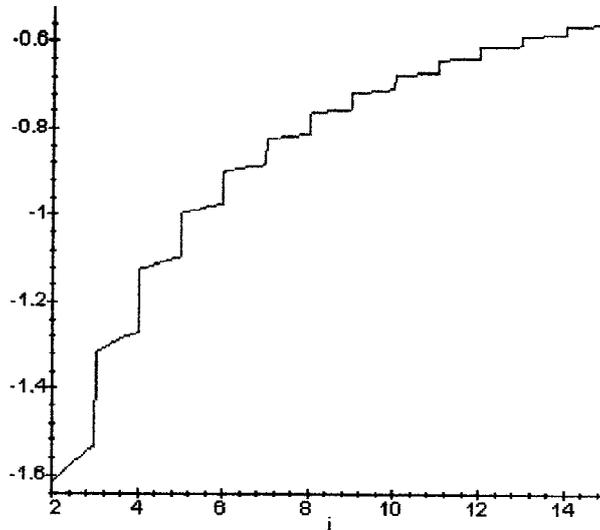


Grappe des points de coupe optimaux  $m_t$  en fonction de  $t$  pour le Quicksort à  $t$  paliers

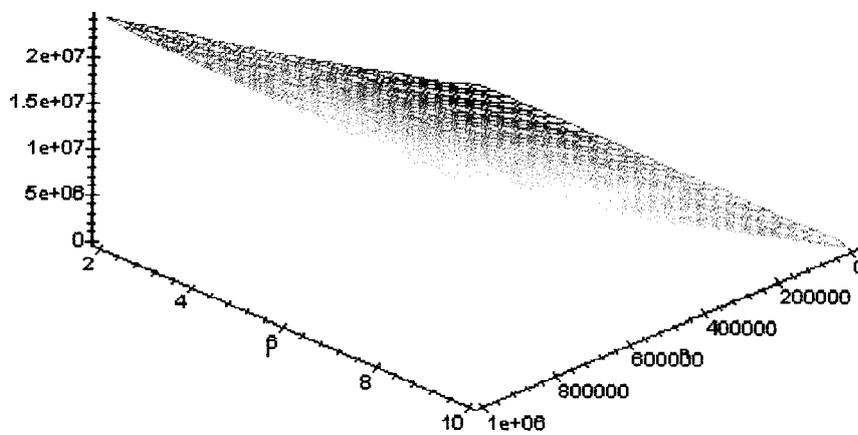


Grappe de  $f_t(m_t)$  avec  $m_t$  optimal en fonction de  $t$  pour le Quicksort à  $t$  paliers

### GRAPHE VIII



Graphe de  $f_i(m_i)$  cumulé avec  $m_i$  optimal en fonction de  $t$  pour le Quicksort à  $t$  paliers



Graphe de  $C_n$  le nombre de comparaisons en fonction de  $n$  et de  $t$  pour le Quicksort à  $t$  paliers



## BIBLIOGRAPHIE

### Références sur l'algorithme du Quicksort:

- D.E. Knuth "Sorting and searching" 1973  
R. Sedgwick "Quicksort" 1980  
"Implementing Quicksort Programs" 1978  
C. Martinez, S. Roura "Optimal sampling for Quicksort" 2000

### Références sur les algorithmes de sélection:

- D.E. Knuth "Sorting and searching" 1973  
D. Doret, U. Zwick "Selecting the median" 1999

### Les documents que nous avons utilisé pour notre travail de recherche:

- D.E. Knuth "Sorting and searching" 1973  
R. Sedgwick, P. Flajolet "Introduction à l'analyse des algorithmes" 1996  
T. Cormen, C. Leiserson et R. Rivest "Introduction à l'algorithmique" 1994  
Thèse de P. Hennequin, 1991