$a=3 \Rightarrow$ a : nbr d'appel recursif dans le pire des cas

1) appel : $\cancel{g}+\frac{d-g+1}{2}-\cancel{1}-\cancel{g}+\cancel{1}=\frac{n}{2}$

2) appel ; $d-g-\frac{d-g+1}{2}+1=\frac{d-g+1}{2}=\frac{n}{2}$

3) $\cancel{g}+3\frac{d-g+1}{4}-\cancel{g}-\frac{d-g+1}{4}-\cancel{1}+\cancel{1}=\frac{n}{2}$

Donc $b=2$

Test cas 1 : $\log_b a=\log_2 3 \iff 1<\log_2 3=\frac{\ln(3)}{\ln(2)}<2$ //pas besoins de savoir plus

$1<1.58-\varepsilon \Rightarrow$ ε val cherché.

si F $\Rightarrow$ complexité $3n+1$

si $g==d+1$ alors

| $T[g]=T[g]+T[d]$

si $d>g+1$ alors

| Algo($T, g, \frac{d-g+1}{2^{\circledast}}-1$)
| Algo($T, g+\frac{d-g+1}{2^{\circledast}}, d$)
| Algo($T, g+\frac{d-g+1}{4^{\circledast}}, g+3\frac{d-g+1}{4}$)
| $F(T, g, d)$

si F $\Rightarrow$ complexité n^2+4

1) $g+\frac{d-g+1}{3}-1-g+1=\frac{n}{3}$

2) $g+2\frac{d-g+1}{3}-1-g-\frac{d-g+1}{3}+1=\frac{n}{3}$

3) $d-g-2\frac{d-g+1}{3}+1=\frac{n}{3}$

$a=3$

$b=3$

//Si appel pas de même taille, impossible appelé master théorème

$\circledast$ 2 est remplacé par 3 ici

Cas 3 car cas 1 et 2 n'ont pas marché : n^2+4 on check que $n^2>n^{1+\varepsilon}$ ✓

(3) $3((\frac{n}{3})^3+4)>c.(n^2+4)$

$c=\frac{1}{2}$ et —

$\frac{n^2+4}{3}$ c $.(n^2+4)$

$a = 4$

$b = 3$

1) $g + \frac{d - g + 1}{3} - 1 - g + 1 = \frac{n}{3}$

4) $g + 7 \frac{d - g + 1}{12} - g - \frac{d - g + 1}{4} - 1 + 1 = \frac{n}{3}$

1) $F() = 2n + 4$

CAS 1 $\quad \Theta(n^{\log \frac{4}{3}})$

remplace 2.

3, 3, 3, 3, 4, 12

2) $F() = n^{\log_3 4} + 5$

CAS 2

$\Theta((\log n) \cdot n^{\log_3 4})$

Recherche du $k^{ème}$ élément :

Médiane d'un tableau   si pair • ⊙ • •

si impair • • ⊙ • •

Le $k^{ème}$ élément : $\underbrace{k-1}_{\text{nbr el plus petit}} < \underset{\text{valeur}}{k} < \underbrace{n-k}_{\text{nbr d'élément} > k}$

Algo naïf 1 : rechercher tt les min et avoir le nbr → prendre le nbr avec médian → $n \times k$

$n \times k - k$

naïf 2 : trier ($n\log n$) → retourner $k^{ème}$ élément. (mais réalise trop d'opérations pour retourner que le $k^{ème}$ élément)

— : quickselect, comme le quick sort mais sans 2 appel recu juste 1.

utilise un pivot (temps linéaire)

pivot

↓

6 3 10 12 14 20 16 30 18 50

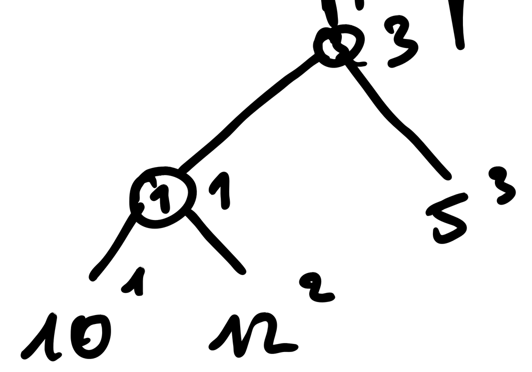
Meilleur Algo : Select-opt.

select-opt ⇒ si tab > 50 ⇝ faire x bloques de 5 et prendre leurs médians ↺ rec
→ puis appliquer ~~pivo~~ quickselect.

$C(n) = \Theta(n)$

Algos arbres min : (arbre binaire parfait)

l'avantage permet lors suppression de val → ne pas tout recalculer



// Dans les noeuds mettre meilleure feuille inférieure ex :

ex > et tas $O(n + k \log(n))$

nbr comp quickselect meilleur dans la pratique