

Recherche du k-ème élément

le problème du «k^ee»

Données:

- Un tableau **T** de taille $n \geq 1$
- Des éléments tous distincts deux à deux
- Un entier **k**

Résultat:

L'élément x de T tel que:

$$|\{y \in T \mid y < x\}| = k-1$$

$$|\{y \in T \mid y > x\}| = n-k$$

dimanche 6 octobre 2013

dimanche 6 octobre 2013

la médiane

Données:

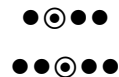
- Un tableau **T** de taille $n \geq 1$
- Des éléments tous distincts deux à deux

Résultat:

l'élément x de T tel que:

$$|\{y \in T \mid y \leq x\}| = \lceil n/2 \rceil$$

$$|\{y \in T \mid y > x\}| = \lfloor n/2 \rfloor$$



dimanche 6 octobre 2013

5 algorithmes

- Algo «naif»
- Algo «naif 2»
- Algo «select» (quickselect»)
- Algo «select opt»
- Algo par arbre min

dimanche 6 octobre 2013

|T| = n

Algo naif

```
def algonatif(T,k) :  
    if k > n : erreur  
    for i = 1..k :  
        indmin = indiceMin(T)  
        if i < k : T[indmin] =  
    return T[indmin]
```

dimanche 6 octobre 2013

|T| = n

Algo naif

```
def algonatif(T,k) :  
    if k > n : erreur  
    for i = 1..k :  
        indmin = indiceMin(T)  
        if i < k : T[indmin] =  
    return T[indmin]
```

n-1 comparaisons

Algo naif 2

- Trier $T[1..n]$
- retourner $T[k]$

dimanche 6 octobre 2013

Algo naif 2

- Trier $T[1..n]$
- retourner $T[k]$

Complexité: $O(n \log n)$

Algorithme **select**

inv: $k \in \{1, \dots, bd - bg + 1\}$
et $bg \leq bd$

```
def select(T, k, bg, bd) :
    indp = indicepivot(T, bg, bd)
    rangpivot = indp - bg + 1
    if (k < rangpivot) :
        return select(T, k, bg, indp - 1)
    elif (k > rangpivot) :
        return select(T, k - rangpivot, indp + 1, bd)
    else :
        return T[indp]
```

Exercice: vérifier l'invariant...

dimanche 6 octobre 2013

Pivotage...

```
def indicepivot(T, bg, bd) :
    p = T[bg]
    l = bg + 1
    r = bd
    while (l <= r) :
        while (l <= bd) and (T[l] <= p) : l += 1
        while (T[r] > p) : r -= 1
        if (l < r) :
            T[r], T[l] = T[l], T[r]
            l, r = l + 1, r - 1
    T[r], T[bg] = p, T[r]
    return r
```

dimanche 6 octobre 2013

Algorithme **select**

inv: $k \in \{1, \dots, bd - bg + 1\}$
et $bg \leq bd$

```
def select(T, k, bg, bd) :
    indp = indicepivot(T, bg, bd)
    rangpivot = indp - bg + 1 → rang du pivot dans T[bg..bd]
    if (k < rangpivot) :
        return select(T, k, bg, indp - 1)
    elif (k > rangpivot) :
        return select(T, k - rangpivot, indp + 1, bd)
    else :
        return T[indp]
```

Exercice: vérifier l'invariant...

dimanche 6 octobre 2013

Pivotage...

```
def indicepivot(T, bg, bd) :
    p = T[bg]
    l = bg + 1
    r = bd
    while (l <= r) :
        while (l <= bd) and (T[l] <= p) : l += 1
        while (T[r] > p) : r -= 1
        if (l < r) :
            T[r], T[l] = T[l], T[r]
            l, r = l + 1, r - 1
    T[r], T[bg] = p, T[r]
    return r
```

pivot = premier élément

Algorithme **select**

```
def select(T,k,bg,bd) :  
    indp = indicepivot(T,bg,bd)  
    rangpivot = indp-bg+1  
    if (k<rangpivot) :  
        return select(T,k,bg,indp-1)  
    elif (k>rangpivot) :  
        return select(T,k-rangpivot,indp+1,bd)  
    else :  
        return T[indp]
```

dimanche 6 octobre 2013

Algorithme «**Select-opt**»

dimanche 6 octobre 2013

Algorithme **select**

```
def select(T,k,bg,bd) :  
    indp = indicepivot(T,bg,bd)  
    rangpivot = indp-bg+1  
    if (k<rangpivot) :  
        return select(T,k,bg,indp-1)  
    elif (k>rangpivot) :  
        return select(T,k-rangpivot,indp+1,bd)  
    else :  
        return T[indp]
```

Complexité: **$O(n^2)$**

cas pire: recherche du plus petit élé^t dans T
trié dans l'ordre décroissant.

dimanche 6 octobre 2013

Algo «select-opt»

(proposé par Blum, Floyd, Pratt, Rivest et Tarjan en 1973)

- Si n est petit, utiliser l'algo **select**.
- Sinon:
 - diviser les n éléments en $\lfloor n/5 \rfloor$ blocs B_i de 5 éléments.
 - **trouver le médian** m_i de chaque B_i
 - **trouver le médian** M des m_i
 - **pivoter** avec M comme pivot
 - continuer dans la bonne partie du tableau (comme dans **select**)...

Idée: garantir que la prochaine étape se fera sur
une **fraction** du tableau initial...

dimanche 6 octobre 2013

Algo «select-opt»

```
def selectopt(T,k,bg,bd) :  
    n = bd - bg + 1  
    if (n < 50) : return select(T,k,bg,bd)[0]  
  
    Taux = [T[IndexMedianQuintuplet(T,bg+j*5)] for j = 1... ⌊n/5⌋ ]  
  
    m = selectopt(Taux, ⌈|Taux|/2⌉ )  
    indp = indicepivotfixe(T,m,bg,bd)  
    rangpivot = indp-bg+1  
  
    if (k < rangpivot) : return selectopt(T,k,bg,indp-1)  
    elif (k > rangpivot) : return selectopt(T,k-rangpivot,indp+1,bd)  
    else :  
        return T[indp]
```

dimanche 6 octobre 2013

Algo «select-opt»

```
def selectopt(T,k,bg,bd) :  
    n = bd - bg + 1  
    if (n < 50) : return select(T,k,bg,bd)[0]  
  
    Taux = [T[IndexMedianQuintuplet(T,bg+j*5)] for j = 1... ⌊n/5⌋ ]  
  
    m = selectopt(Taux, ⌈|Taux|/2⌉ )  
    indp = indicepivotfixe(T,m,bg,bd)  
    rangpivot = indp-bg+1  
  
    if (k < rangpivot) : return selectopt(T,k,bg,indp-1)  
    elif (k > rangpivot) : return selectopt(T,k-rangpivot,indp+1,bd)  
    else :  
        return T[indp]
```

l'ind. du médian de

retourne

Algo «select-opt»

```
def selectopt(T,k,bg,bd) :  
    n = bd - bg + 1  
    if (n < 50) : return select(T,k,bg,bd)[0]  
  
    Taux = [T[IndexMedianQuintuplet(T,bg+j*5)] for j = 1... ⌊n/5⌋ ]  
  
    m = selectopt(Taux, ⌈|Taux|/2⌉ )  
    indp = indicepivotfixe(T,m,bg,bd)  
    rangpivot = indp-bg+1  
  
    if (k < rangpivot) : return selectopt(T,k,bg,indp-1)  
    elif (k > rangpivot) : return selectopt(T,k-rangpivot,indp+1,bd)  
    else :  
        return T[indp]
```

l'ind. du médian de

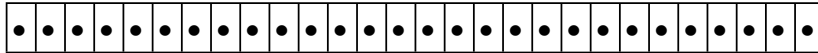
retourne

dimanche 6 octobre 2013

Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

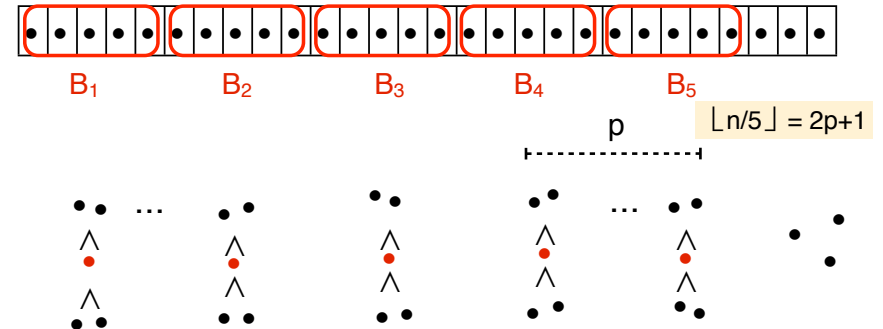
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



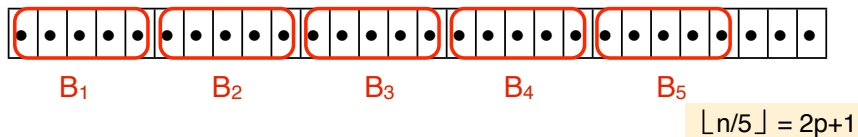
dimanche 6 octobre 2013

dimanche 6 octobre 2013

Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?

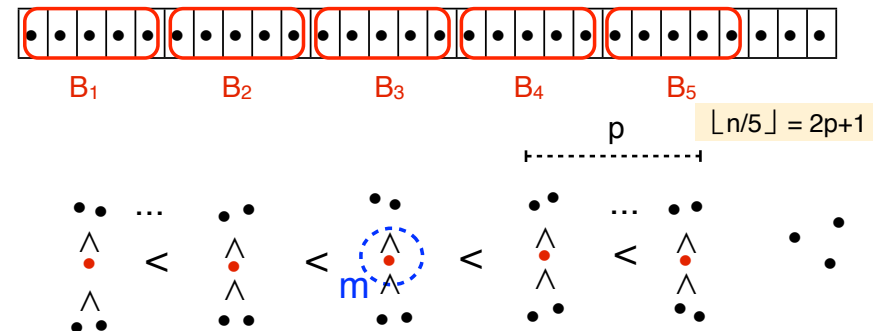


dimanche 6 octobre 2013

Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?

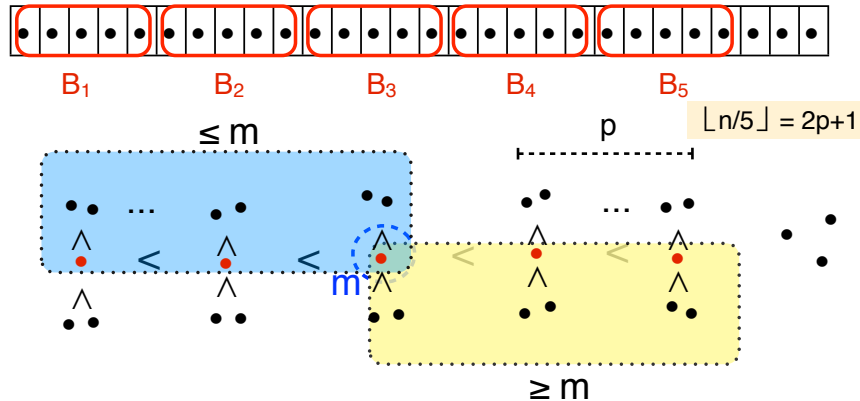


dimanche 6 octobre 2013

Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

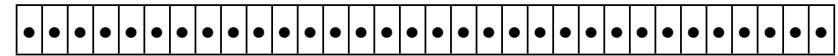
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

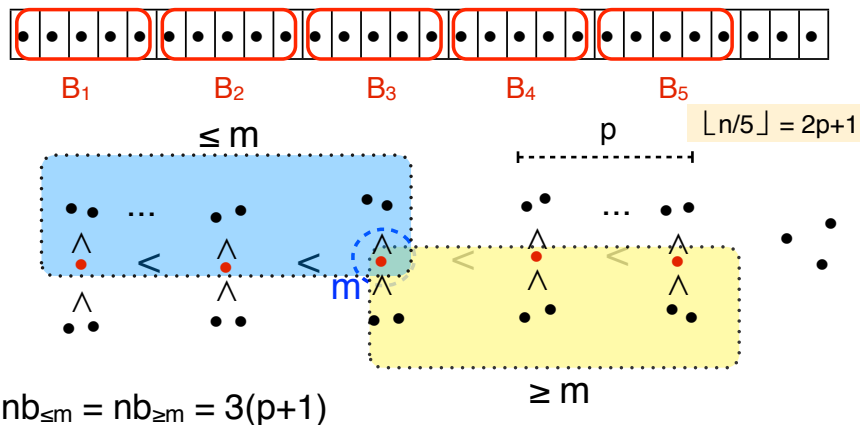
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

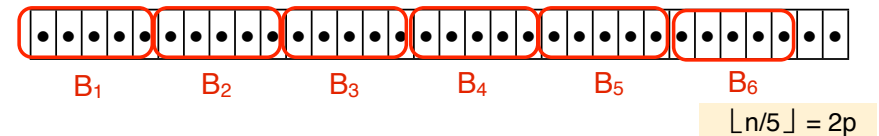
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

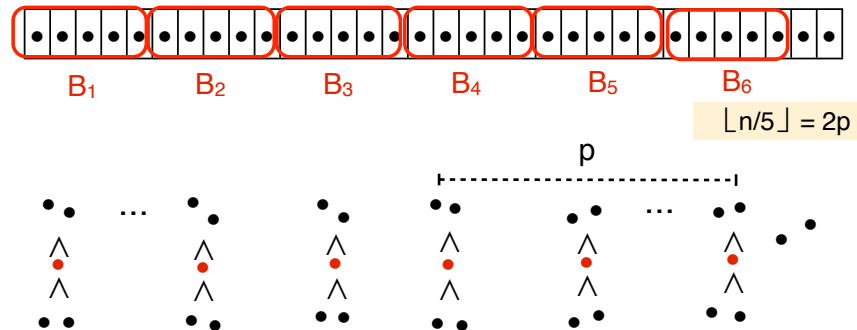
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

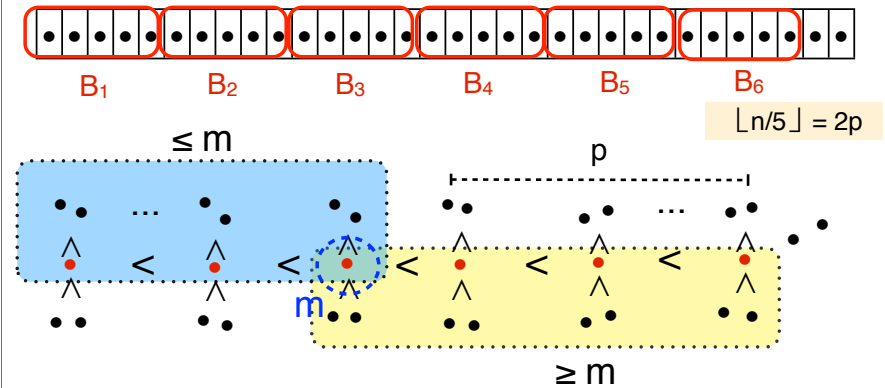
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

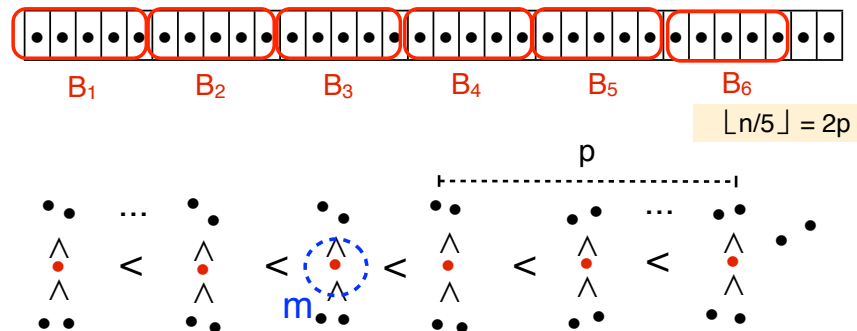
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

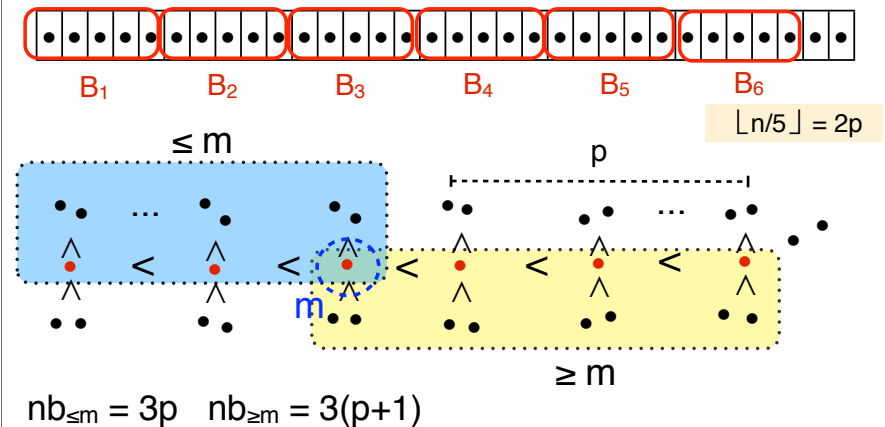
Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

La complexité de l'algorithme est bonne car le pivot m choisi n'est jamais «trop mauvais»...

Comment évaluer le nb d'éléts $\leq m$ et $\geq m$?



Algo «select-opt» - complexité

1) $\lfloor n/5 \rfloor = 2p$, $nb_{\leq m} = 3p$ $nb_{\geq m} = 3(p+1)$

2) $\lfloor n/5 \rfloor = 2p+1$, $nb_{\leq m} = nb_{\geq m} = 3(p+1)$

dimanche 6 octobre 2013

Algo «select-opt» - complexité

1) $\lfloor n/5 \rfloor = 2p$, $nb_{\leq m} = 3p$ $nb_{\geq m} = 3(p+1)$

$$p = \frac{1}{2} \lfloor n/5 \rfloor = \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

$$p+1 = \lceil \frac{1}{2}(2p+1) \rceil = \lceil \frac{1}{2}(\lfloor n/5 \rfloor + 1) \rceil$$

2) $\lfloor n/5 \rfloor = 2p+1$, $nb_{\leq m} = nb_{\geq m} = 3(p+1)$

$$p+1 = \lceil \frac{1}{2}(\lfloor n/5 \rfloor) \rceil$$

$$p+1 = \frac{1}{2}(\lfloor n/5 \rfloor + 1) = \lceil \frac{1}{2}(\lfloor n/5 \rfloor + 1) \rceil$$

dimanche 6 octobre 2013

Algo «select-opt» - complexité

1) $\lfloor n/5 \rfloor = 2p$, $nb_{\leq m} = 3p$ $nb_{\geq m} = 3(p+1)$

$$p = \frac{1}{2} \lfloor n/5 \rfloor = \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

$$p+1 = \lceil \frac{1}{2}(2p+1) \rceil = \lceil \frac{1}{2}(\lfloor n/5 \rfloor + 1) \rceil$$

2) $\lfloor n/5 \rfloor = 2p+1$, $nb_{\leq m} = nb_{\geq m} = 3(p+1)$

dimanche 6 octobre 2013

Algo «select-opt» - complexité

1) $\lfloor n/5 \rfloor = 2p$, $nb_{\leq m} = 3p$ $nb_{\geq m} = 3(p+1)$

$$p = \frac{1}{2} \lfloor n/5 \rfloor = \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

$$p+1 = \lceil \frac{1}{2}(2p+1) \rceil = \lceil \frac{1}{2}(\lfloor n/5 \rfloor + 1) \rceil$$

2) $\lfloor n/5 \rfloor = 2p+1$, $nb_{\leq m} = nb_{\geq m} = 3(p+1)$

$$p+1 = \lceil \frac{1}{2}(\lfloor n/5 \rfloor) \rceil$$

$$p+1 = \frac{1}{2}(\lfloor n/5 \rfloor + 1) = \lceil \frac{1}{2}(\lfloor n/5 \rfloor + 1) \rceil$$

Conclusion:

$$nb_{\leq m} = 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

$$nb_{\geq m} = 3 \lceil \frac{1}{2}(\lfloor n/5 \rfloor + 1) \rceil \geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

dimanche 6 octobre 2013

Algo «select-opt» - complexité

$$nb_{\leq m}, nb_{\geq m} \geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

comme $\lfloor n/5 \rfloor \geq (n-4)/5$, on a:

$$\begin{aligned} nb_{\leq m}, nb_{\geq m} &\geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil \geq 3/2 \cdot \lfloor n/5 \rfloor \geq (3n-12)/10 \\ &\geq 3n/10 - 2 \end{aligned}$$

Algo «select-opt» - complexité

$$nb_{\leq m}, nb_{\geq m} \geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

comme $\lfloor n/5 \rfloor \geq (n-4)/5$, on a:

$$\begin{aligned} nb_{\leq m}, nb_{\geq m} &\geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil \geq 3/2 \cdot \lfloor n/5 \rfloor \geq (3n-12)/10 \\ &\geq 3n/10 - 2 \end{aligned}$$

Les appels récursifs se font donc sur des sous-tableaux de taille $\leq n-3n/10+2$, donc $\leq 7n/10+2$.

$$C(n) = C(\lfloor n/5 \rfloor) + C(7n/10+2) + O(n)$$

dimanche 6 octobre 2013

dimanche 6 octobre 2013

Algo «select-opt» - complexité

$$nb_{\leq m}, nb_{\geq m} \geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil$$

comme $\lfloor n/5 \rfloor \geq (n-4)/5$, on a:

$$\begin{aligned} nb_{\leq m}, nb_{\geq m} &\geq 3 \lceil \frac{1}{2} \lfloor n/5 \rfloor \rceil \geq 3/2 \cdot \lfloor n/5 \rfloor \geq (3n-12)/10 \\ &\geq 3n/10 - 2 \end{aligned}$$

Les appels récursifs se font donc sur des sous-tableaux de taille $\leq n-3n/10+2$, donc $\leq 7n/10+2$.

Algo «select-opt» - complexité

Proposition:

Si $t(n) = \sum_i a_i t(n/b_i) + c \cdot n$ avec $\sum_i a_i/b_i < 1$
alors $t(n) = \Theta(n)$

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

L'algorithme select-opt est en temps linéaire !

dimanche 6 octobre 2013

dimanche 6 octobre 2013

Algo «select-opt» - complexité

$$C(n) = C(\lfloor n/5 \rfloor) + C(7n/10+2) + O(n)$$

Proposition:

Si $t(n) = \sum_i a_i t(n/b_i) + c.n$ avec $\sum_i a_i/b_i < 1$
alors $t(n) = \Theta(n)$

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

L'algorithme select-opt est en temps linéaire !