Corrigé type Examen de Algorithmique Avancée et Complexité

Année 2020/2021 Par Mr Hanich

Exercice 1

Partie 1

$$T_1(n) = \sum_{i=1}^{n} (1+2i) + 2 = n + 2 + 2\sum_{i=1}^{n} i$$

(2)

$$= n + 2 + 2\,(n+1)\,n/2 = n + 2 + n^2 + n$$

$$= n^2 + 2n + 2 = \underline{O(n^2)}$$

Partie 2

$$T_2(n) = \sum_{i=10}^{n-10} (1 + (i+10 - (i-10) + 1) * 2) + 2$$

(2)

$$= 2 + \sum_{i=10}^{n-10} (1 + 21 * 2) = 2 + \sum_{i=10}^{n-10} 43$$

$$= 2 + (n - 10 - 10 + 1) * 43$$

$$= 43n - 861 = \underline{O(n)}$$

Partie 3

$$T_3(n) = 1 + \sum_{i} ((i+1) + 2 + 1) + 1 \quad \text{avec } i = n, \frac{n}{2}, \frac{n}{4}, \dots, \frac{n}{2^{\log_2 n}}$$

$$= 2 + \sum_{i} 3 + 2\sum_{i} i \quad \text{avec } i = n, \frac{n}{2}, \frac{n}{4}, \dots, 1$$

(2)

$$= 2 + 3\log_2 n + 2\underbrace{\left(n + \frac{n}{2} + \frac{n}{4} + \dots + \frac{n}{2^{\log_2 n}}\right)}_{S}$$

le nombre de valeurs $= \log_2 n$

calcul de S

nous avons $S - \frac{1}{2}S = n - \frac{1}{2}$ (car $\frac{1}{2}S = \frac{n}{2} + \frac{n}{4} + \dots + \frac{1}{2}$)

$\Rightarrow \frac{1}{2}S = n - \frac{1}{2} \Rightarrow S = 2n - 1$

donc $T_3(n) = 2 + 3\log_2 n + 2(2n-1)$

$= 4n + 3\log n = \underline{O(n)}$

Corrigé complexité examen 2020-2021

M. Hamiche IF

Exercice n°2

L'Algorithme se base sur la sélection du fichier de taille inférieur parmi les fichiers non choisi, et cela dans chaque itérations du processus glouton.

La préparation de donnée implique :

1- une initialisation du tableau S à zero

2- tri croissant, selon la taille des fichiers, du tableau P contenant les information sur les fichier, l'indice du tableau P sera utilisé comme identifiant du fichier de ce numéro.

```
Algorithme
Const n = ..
Var
    P : tableau [1..n] réel;
    S : tableau [1..n] entier;
    L, Som : réel;
Début
    <tri du tableau P>      { tri par fusion par exemple O(n log n)

    <initialisation de S>
        Pour i := 1 à n faire S[i] := 0 fait      O(n)
    lire(L);
    Som := 0; i := 1;                             O(1)
    tant que Som + P[i] < L                       le pire des cas correspond à l'ajout de tout les fichier càd nbr d'itération = n
    faire                                         donc dans ce cas O(n)
        S[i] := 1;
        i := i + 1;
        Som := Som + P[i]
    fait;
Fin.
```

la complexité totale

DM Prod - Suite
Mr Hamiche F

remarque

si le tableau n'est pas trié on aura une complexité $O(n^2)$

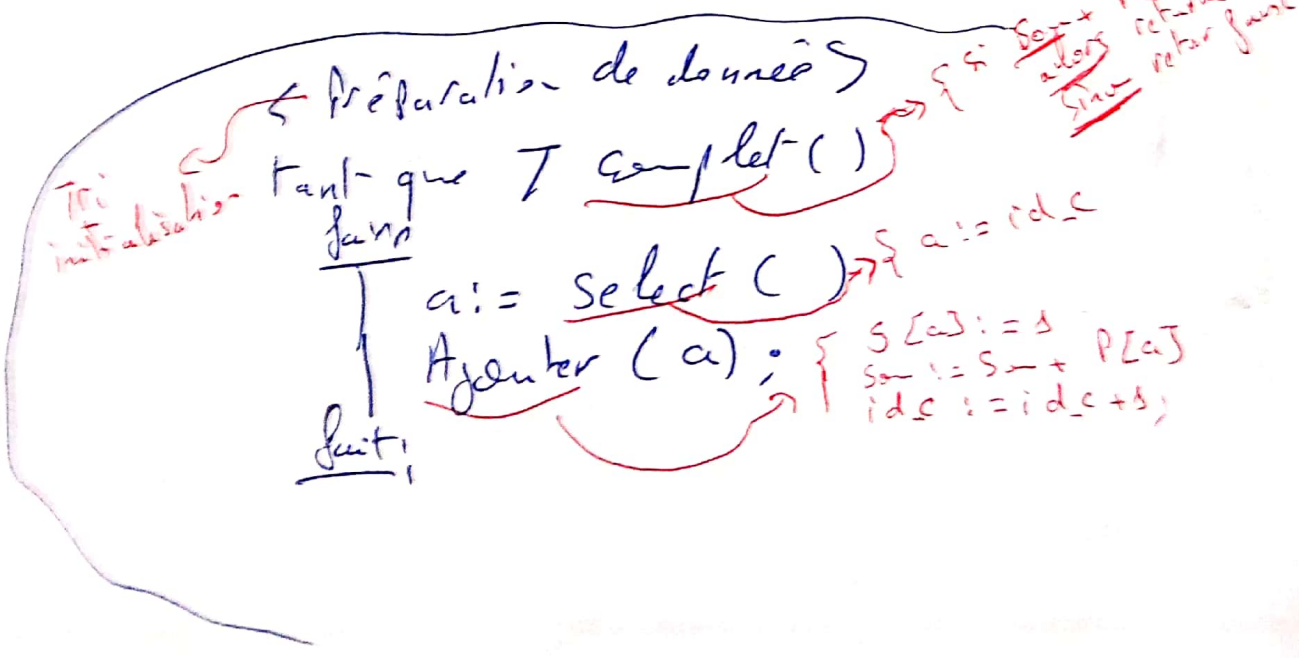
car dans la boucle pour chaque itération on aura besoin de rechercher le min dans le tableau qui coutera $an+b$ et donc

$$T(A) = A \times (an+b) = O(n^2)$$

A : nbr d'itération ; $(an+b)$: cout de la recherche

il est possible de donné l'algorithme sous le schéma Vue en cours (mais puisque ce n'est pas compliqué, on a donnée cette version simplifier)

schéma Vue en cours



c) la version optimiser de l'ordre de $O(n \log n)$
Va se basé sur la technique diviser pour régner
et l'Algorithme utilise des fonction récursif vue
en cours
Algorithme Rech_Som_x

3/5

```
Début
    Tri_fusion (T, n) ;   ~ O(n log n)
    i := 1 ;
    tant que i <= n et Tr
    faire
        b := x - T[i] ;
        tr := Rech_dechotomique (T, i, n, b) ;
        i := i + 1 ;
    fait ;
fin.
```

$O(n \log n)$

donc

~~$T(n)$~~ l'Algorithme est de l'ordre de $O(n \log n) + O(n \log n) = O(n \log n)$

Recherche déchotomique de b dans le tableau T entre la case i et la case n
de l'ordre de $O(\log n)$

$O(\log n + n \log n)$

Exercice 3

1- Un algorithme naïve pour ce problème fait la recherche sur le tableau brut sans prétraitement à l'avance, et se base sur une recherche très simple de deux indices de table i, j tq T[i] + T[j] = x

Algorithme Rech_Som_x

Début
  Lire (x)
  ~~Pour i := 1~~
  i := 1 ;
  tant que i <= n et ~T tr
  faire
    j := i ;
    b := x - T[i]
    tant que j <= n et ~T tr
    faire
      si T[j] = b alors écrire (i, j)
        tr := vrais
      sinon j := j + 1
    fsi ;
    fait ;
    fait ; i := i + 1
  fin.

$$T(n) = 2 + \sum_{i=1}^{n} \left(2 + 5 + (n-i) \times 3 \right)$$

$$= 2 + 3n^2 + 7n - 3 \sum_{i=1}^{n} i = 3n^2 + 7n + 2 - \frac{3}{2}(n^2+n)$$

$$= \frac{3}{2}n^2 + \frac{11}{2}n + 2 = O(n^2)$$