

Examen - UV A31 Algorithmme

30 mars 2017

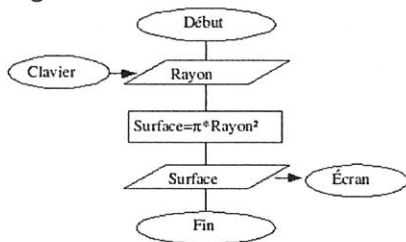
Question 1 (4 points)

- (1) Donner 5 classes de complexités, les plus connues, n étant la taille des données.
- (2) Quelles sont les classes critiques de complexité ?

Question 2 (6 points)

Donner la complexité temporelle des 3 algorithmes ci-dessous :

(a) Algorithme surface d'un cercle



(b) Algorithme puissance d'un nombre

```
Fonction puissance1 (x: réel, n: entier): réel
Res <- 1;
Tant que n > 1
faire
    res <- res * x;
    n <- n - 1;
Fin faire
Retourner res
```

(c) Algorithme de recherche dichotomique

```
int RechercheDichotomique (int A[], int First, int Last, int Value)
Début algorithme
    int Min, Max, Mid;
    Min = First ; Max = Last;
    Tant que (Min ≤ Max)
    Faire
        Mid = (Min+Max)/2;
        Si (A[Mid] = x) alors return (Mid); Fin si
        Si (A[Mid] < x) alors Min = Mid + 1; Sinon max = Mid - 1; Fin Si
    Fin faire
    Return(-1);
Fin algorithme
```

Examen - UV A31 Algorithmme

30 mars 2017

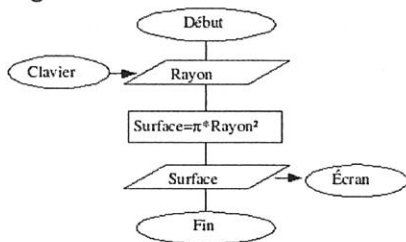
Question 1 (4 points)

- (1) Donner 5 classes de complexités, les plus connues, n étant la taille des données.
- (2) Quelles sont les classes critiques de complexité ?

Question 2 (6 points)

Donner la complexité temporelle des 3 algorithmes ci-dessous :

(a) Algorithme surface d'un cercle



(b) Algorithme puissance d'un nombre

```
Fonction puissance1 (x: réel, n: entier): réel
Res <- 1;
Tant que n > 1
faire
    res <- res * x;
    n <- n - 1;
Fin faire
Retourner res
```

(c) Algorithme de recherche dichotomique

```
int RechercheDichotomique (int A[], int First, int Last, int Value)
Début algorithme
    int Min, Max, Mid;
    Min = First ; Max = Last;
    Tant que (Min ≤ Max)
    Faire
        Mid = (Min+Max)/2;
        Si (A[Mid] = x) alors return (Mid); Fin si
        Si (A[Mid] < x) alors Min = Mid + 1; Sinon max = Mid - 1; Fin Si
    Fin faire
    Return(-1);
Fin algorithme
```

Question 3 (4 points)

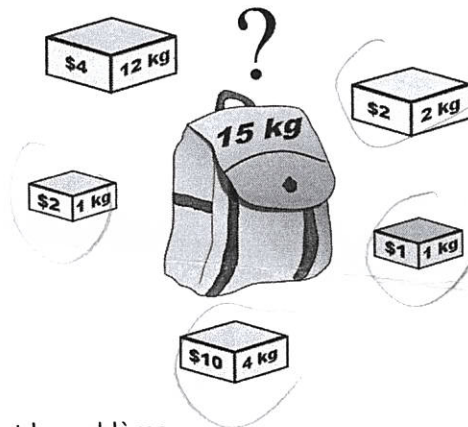
Soit la relation de récurrence de la fonction $W(i, j)$ définie par :

$$\begin{cases} 1 & \text{si } j = 0 \\ 1 & \text{si } j = i \\ 0 & \text{si } j > i \\ W(i-1, j-1) + W(i-1, j) & \text{si } 0 < j < i \end{cases}$$

- (1) Donnez la fonction récursive en C ou pseudo C.
- (2) Donnez une trace sous la forme d'un arbre du déroulement de l'appel $W(4,2)$
- (3) Estimer la complexité de cette fonction.

Question 4 (4 points)

Soit le problème de sac à dos. Quelles boîtes choisir afin de maximiser la somme emportée tout en ne dépassant pas les 15 kg autorisés



- (1) Modéliser mathématiquement le problème
- (2) Quelle est la complexité de l'algorithme ?
- (3) Développer l'algorithme

Question 5 (2 points)

Développer un algorithme de recherche d'un entier dans un arbre binaire.

Nous suggérons l'utilisation des fonctions suivantes :

- Boolean Recherche (Arbre arbre, int x) ✗
Retourne vrai si x est dans l'arbre et faux sinon
- Boolean EstVide(Arbre arbre) ✗
Retourne vrai si arbre est vide et faux sinon
- Arbre FilsGauche(Arbre arbre)
Retourne la racine du sous arbre gauche
- Arbre FilsDroit(Arbre arbre)
Retourne la racine du sous arbre droit