SAC À DOS AVEC L'APPROCHE GLOUTONNE

October 25, 2020

1 Description du probleme

"Etant donné plusieurs objets possédant chacun un poids et une valeur et étant donné un poids maximum pour le sac, quels objets faut-il mettre dans le sac de manière à maximiser la valeur totale sans dépaser le poids maximal autorisé par le sac?"
Le but est de remplir le sac sans dépasser la limite de poids maximum avec la plus grande valeur totale et le plus rapidement possible.
Sac de poids maximal w et n objets. Pour cgaque objet i, nous avons un poids Wi et une valeur Pi. Xi est associé à l'objet i de sorte que xi=1 si l'objet i est mis dans le sac et vaut 0 sinon.

2 Algorithme

Algorithme SacADos

```
        Type Objets = tableau[1....4,1....n] de réels
        var  O:Objets; Wo,Wmax i,n:entiers
       // ce tableau contient 4 lignes dont la première O[1,i]contient les valeurs des objets
       //la deuxième ligne O[2,i]contient le poids des objets
      //la troisième O[3,i]ligne contiendra le rapport valeur/ poids ou bien pi/Wi
     //la dernière ligne O[4,i] se verra attribuer des 0(dans le cas où on met l'objet dans le sac) ou des 1(si
```

Debut

```
  pour i allant de 1 à n faire
      O[3,i] ← O[1,i]/O[2,i]  //O[1,i] représente pi et O[2,i] représente wi
  finpour
  TriDecroissant(O) //tri décroissant des objets suivant le rapport pi/wi
  Wo ← 0
  pour i allant de 1 à n faire
    si O[2,i]+ Wo <= Wmax alors
        O[4,i] ←  1 //c'est à dire que l'objet est mis dans le sac
        Wo ←  Wo+O[2,i]
    sinon
          O[4,i] ←  0 //l'objet n'est pas dans le sac
    finsi
 finpour
```

Fin

procedure

Procedure TriDecroissant(T:tableau [1...n,1...m])

```
     var x,j,i,m:entiers
```

Debut

```
     m ← length(T)
     pour i allant de 2 à m-1 faire
          x ←  T[3,i]
          j ←  i-1
     tant que j > 0 et T[3,j] < x faire
          T[1,j+1] ←  T[1,j]
          T[2,j+1] ←  T[2,j]
          T[3,j+1] ←  T[3,j]
          j ←  j-1
     fintantque
          T[3,k+1] ←  x
     finpour
```

Finprocedure

3 Complexite

complexité du tri(Ttri):$O(n^2)$
$T = n(1) + Ttri + 1 + n(1 + 2)$
$= Ttri + 4n + 1$
$= O(n^2)$