Université MB de M'sila
Faculté de MI
Département d'informatique
Formation Master 1 IDO

Module Méthodes d'optimisation
Examen session normale janvier 2024
Durée : 1 heure 30 minutes

Exercice 1 (questions de cours) (06 points)

1) **Problème du cycle eulérien ; P**
   Problème du cycle hamiltonien ; NP-C
   Problème du plus court chemin ; P
   SAT-2 (sous FNC) ; P
   SAT-3 (sous FNC) ; NP-C
   SAT (sous FND) ; P...3
2) **Parce que SAT-k (k>3) $\leq_P$ SAT-3** 0.5
3) **Pour DP : les sous pbs sont imbriqués, pour B&B, les sous pbs sont disjoints**...........................1
4) **- Diviser le pb en sous pbs disjoints**
   - Une heuristique pour calculer une borne faisable
   - Relaxer les contraintes pour calculer une borne non faisable...1.5

Exercice 2 (05 points)

1) **B n'existe pas**...0.5
2) **Calcule les rangs des éléments du tableau**...1
3) **O(n^2)** 0.5
4) **Algorithme plus efficace** 2.5

```
Int[] A(int[] B; int n) {
      for(i = 0 ;i < n ;i++)
            {C[0][i] = B[i] ;
             C[1][i] = i ;      }
      Sort C according first row
      for(i = 0 ;i < n ;i++)
         A[C[1][i]]= i;
      return A[] ;}
```

Complexité =O(nlogn) 0.5

Exercice 3 (09 points)

1) **Soit S le sous ensemble recherche, une solution peut être représentée par un vecteur x de n bits $x_i = 1\ si\ a_i \in S\ et\ 0\ sinon$**...1
2) $\begin{cases} \sum_{i=1}^{n} x_i a_i = M \\ x_i \in \{0,1\} \end{cases}$ 1
3) 2^n...0.5
4) **Prendre les entiers par ordre décroissant.**

```
Int[] x(int[] a; int n, int M) {
      Sor a[] in descending order
      s=0;
      for(j = i ; i < n ; i++)
          if (s+ a[i] <= M)
               {x[i] = 1 ;   s+=a[i] ; }
          else
                 x[i]= 0 ;
      return x[] ;}
```
…………………………………………………………..………..……….2.5

Pour E = {3, 1, 5, 2, 7} et M = 10 → tri de E ={7,5,3,2,1} → x={1,0,1,0,0}………………….…..……….1

5) **Ce problème peut se réduire à un KSP où v_i = w_i = a_i et capacité du sac = M**……………………2.5

a_i	i	0	1	2	3	4	5	6	7	8	9	10	x
	0	0	0	0	0	0	0	0	0	0	0	0	
3	1	0	0	0	3	3	3	3	3	3	3	3	1
1	2	0	1	1	3	4	4	4	4	4	4	4	0
5	3	0	1	1	3	4	5	6	6	8	9	9	1
2	4	0	1	2	3	4	5	6	7	8	9	10	1
7	5	0	1	2	3	4	5	6	7	8	9	10	0

6) **Cet algorithme glouton est optimal**……………………..……………………………………..…..……….0.5