

Exercice 1 : Questions de cours (03 points)

- 1) Un problème de décision = un problème dont la réponse est oui ou non.0.5
 Exemple= l'entier n est-il premier ?0.5
- 2) $P \subset NP$; $NP \cap NP\text{-Difficile} = NP\text{-Complet}$; $NP\text{-Complet} \subset NP$1
- 3) Si le problème est décomposable en sous problèmes qui chevauchent dont la solution est une suite de décisions satisfaisant l'hypothèse d'optimalité de Bellman.1

Exercice 2 : (05 points)

	Temps d'exécution pour une instance de taille 10	Temps d'exécution pour une instance de taille 100	Taille maximale de l'instance traitée en une heure
A1	$n^2/f = 100/f$ s1	$n^2/f = 10000/f$ s.....0.5	$n^2/f = 3600 \Rightarrow n = 60\sqrt{f}$1
A2	$2^n/f = 2^{10}/f$ s.....1	$2^n/f = 2^{100}/f$ s.....0.5	$2^n/f = 3600 \Rightarrow n = \log_2(3600f)$1

Exercice 3 (12 points)

- 1)
$$\begin{cases} \max \sum_{i=1}^n x_i a_i \\ \sum_{i=1}^n x_i a_i \leq B \\ x_i \in \{0, 1\} \end{cases}$$
1.5
- 2) Espace de recherche $S = \{0,1\}^n$; $|S| = 2^n$ 0.5
- 3) Discret (combinatoire).0.5
- 4) Les entiers = les objets ; valeurs des entiers = valeurs des objets = poids des objets = a_i ; la capacité du sac = B.1.5
- 5) Ce problème est une réduction polynomiale du problème du sac à dos.0.25
- 6) Ce problème est NPC puisque le KSP l'est aussi.0.25
- 7)3

ai	i\b	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	2	2	2	2	2	2	2	2	2	2
4	2	0	0	2	2	4	4	6	6	6	6	6	6
3	3	0	0	2	3	4	5	6	7	7	9	9	9
5	4	0	0	2	3	4	5	6	7	8	9	10	11

- 8) void DP(int n , int[] a , int B)3


```

{ for (i=0 ; i<=n ; i++) Tab[i][0]= 0 ; // remplir la première colonne par des 0
  for (b=0 ; b<=B ; b++) Tab[0][b]= 0 ; // remplir la première ligne par des 0
  for (i=0 ; i<=n ; i++) // remplir la table
    for (b=0 ; b<=B ; b++)
      if (a[i] > b ) Tab[i][b]= Tab[i-1][b] ;
      else Tab[i][b]= max (Tab[i-1][b] , a[i]+Tab[i-1][b- a[i]]) ;
  b=B ; // calculer la solution
  for (i=n ; i>0 ; i--)
    if ( Tab[i][b] <> Tab[i-1][b] ) { solution[i]=1 ; b -= a[i] ; }
  return (solution , Tab[n][B]) ; } // output
Complexité temporelle = O(Bn) .....0.5
Complexité spatiale = O(Bn).....0.5
            
```