MB University of M'sila
Faculty of MI
Department of Computer Science
Master 1 IDO

Combinatorial Optimization 1
Normal Session exam, January 2026
Duration : 2 hours
Model answer

Exercise 1 (Course questions) (04 points)

1) **Decision problem:** A problem such that the answer is **yes or no** for a given input.....0.5
2) **Relations between complexity classes :**
   $\mathbf{P \subseteq NP}$; $\mathbf{NP\text{-}complete \subseteq NP}$; $\mathbf{NP\text{-}hard \supseteq NP\text{-}complete}$; $\mathbf{NP \cap NP\text{-}hard = NP\text{-}complete}$.....0.5
3) **Optimal solution x^* : $x^* \in S$ such that $f(x^*) \leq f(x), \forall x \in S$** 0.5
4) **Choice between A1 and A2 : A1 because it is faster.**.....0.5
5) **Memorization table in DP: To store solutions of subproblems and avoid recomputation, reducing time complexity. .**.....1
6) **B&B becomes most efficient when bounds are closes and a good initial feasible solution.** 1

Exercise 2 (06 points)

1) **Length of tour ACDB $= 4 + 4 + 6 + 8 = 22$** 0.5
2) **Number of possible tours $= \frac{(n-1)!}{2}$ For $n = 4$: $\frac{3!}{2} = 3$ tours** 0.5
3) **Nearest Neighbor tour : ADCB Length $= 2 + 4 + 7 + 8 = 21$**.....1.5
   This tour gives an upper bound, because any feasible tour length ≥ optimal tour length.....1.5
4) **Heuristic computing the other bound (Lower Bound)** 2

   Algorithm LowerBound_TSP(G):
   Compute the UB using the heuristic NN above;
   Replace one edge by another shorter

Exercise 3 (10 points)

1) **a) for c = [10 , 5 , 2 , 1] and A= 36: x= [3 , 1 , 0 , 1] and T=5.** 1.5
   b) time: O(n log n + n * (A / min(c))) ≈ O(n * A) ; space : O(n) + O(1) = O(n). 0.5
2) **a) Coin Change**.....0.25
   b) Let $x_i \geq 0$ integers: number of times coin i is used. 2.5

   Minimize $\sum_{i=1}^{n} x_i$

   s.t. $\sum_{i=1}^{n} c_i x_i = A$

   $x_i \in N, i = 1..n$

   c) Search space: $\{(x_1, \ldots, x_n)\} / x_i \in N$ **with** $\sum c_i x_i \leq A$; $x_i \leq \frac{A}{c_i}$; **its size:** $\prod_{i=1}^{n} int\left(\frac{A}{c_i}\right)$.....2
3) **Greedy algorithm.** 0.25
4) **a) dynamic programming algorithm.**

```
DP_CC(c[], A):
  n = length(c)
  for j = 0 to A:
     DP[0][j] = +∞
  DP[0][0] = 0
  for i = 1 to n:
     DP[i][0] = 0
     for j = 1 to A:
        if c[i] > j:
           DP[i][j] = DP[i-1][j]
        else:
           DP[i][j] = min(DP[i-1][a], DP[i][j - c[i]]+1)
        return DP[n][A]
```
.....2

b) $T_{DP} \leq T_1$
because DP is an exact method, greedy algorithm is an approximate method and CC is a minimization problem. 1