CS345: DESIGN AND ANALYSIS OF ALGORITHMS

MID-SEMESTER EXAM – GRADING SCHEME

Question 1 (8 marks)

- *Solution*:
    1. i,
    2. k (or k-1),
    3. k+1 (or k),
    4. j,
    5. low = min(lowL, lowR),
    6. high = max(highL, highR),
    7. profit = max(profitL, profitR, highR-lowL),
    8. Complexity = O(n)
- *Alternate solution*:
    1. i,
    2. k (or k-1),
    3. k+1 (or k),
    4. j,
    5. low = min(A[i], ... ,A[k]),
    6. high = max(A[k+1], ... A[j]),
    7. profit = max(profitR, profitL, high-low),
    8. Complexity = O(nlogn)

- **1 mark** for each correct blank

Question 2a (2 marks)

- *Recursive term:* P(i,v) (**1 mark**),
- *Interpretation:* Profit from boxes b_1 to b_i with volume of bag not exceeding v (**1 mark**)
  Not mentioned which boxes: **-0.5 mark**
  ~~Exact volume v instead of volume ≤v: **-0.5 mark**~~

Question 2b (3 marks)

- *Base case* (**1 mark**):
  $P(0,v) = 0$ for $0 \leq v \leq m$
  If this is missing: **-0.5 mark**
  (or)
  $P(1,v) = 0$ for $0 \leq v < v_i$
  $P(1,v) = p_i$ for $v_i \leq v \leq m$

- *General case:*
  Recursive formulation (**1 mark**):
  $P(i,v) = \max \{ P(i-1,v-v_i)+p_i \text{ if } v_i \leq v$
  $P(i-1,v)$
  Justification (**1 mark**): take or not take i^{th} box.
  Justification missing: **-1 mark**
  Justification incomplete: **-0.5 mark**

Question 2c (3 marks)

- Base case: **1 mark**
- Not take i^{th} box case: **0.5 mark**
- Check of $v_i \leq v$: **0.5 mark**
- Take i^{th} box case: **0.5 mark**
- Final return statement: **0.5 mark**

  Mistake in loop: **-1 mark**
  $+ p_i$ missing in equation: **-0.5 mark**
  Missing base case for $v = 0$ to m: **-0.5 mark**

  Marks given for both top down and bottom up approach.
  Also accepted answers for recursive formulation for bags b_i to b_n, base case must be accordingly modified.

Question 3a (2 marks)

- **2 marks** for a complete and correct definition
- **1 mark** if one of the conditions [$a_i < a_j$ and $i < j$] is missing
- **0 marks** for an incorrect definition or stating the vertices/edges explicitly

Question 3b (2 marks)

- **2 marks** if the graph is drawn as per the definition in **3a**
- **0 marks** if more than 2 edges (or vertices) are missing/wrong

Question 3c (2 marks)

- **2 marks** if the stated problem on the DAG defined in **3a** solves LMS
- **0 marks** otherwise

Question 4 (9 marks)

- **9 marks** for a correct counter-example.
- **0 marks** if the algorithm on the given example outputs an optimal-size subset.

Question 5 (9 marks)

```
if D[s] <> 0:
     D[s] = 0 and return;
else:
     for each edge (u,v) in E:
          if (D[v] > D[u] + weight(u,v)):
               D[v] <- D[u] + weight(u,v)
```

Time complexity is O(m + n).

Any algorithm that exceeds this time complexity will get **0 marks**.

Therefore:

- **No marks** if your algorithm assumes G to be a DAG.
- **No marks** if your algorithm sorts D in increasing or decreasing order.
- **No marks** if your algorithm performs Dijkstra's algorithm.
- **No marks** if your algorithm uses Priority Queue (Binary heap or Fibonacci heap).