QUIZ 2 GRADING SCHEME

Question 1>

- −1 mark for each “least-overlap” job (when multiple such jobs exist) that does not contradict the statement.
- -5 marks: if the least overlap job is present in an optimal solution.

Question 2>

- -1 mark for each incorrect cross edge.(backward edges drawn are also considered as an incorrect cross edge)
- -1 for each extra cross edge(minimum cross edges were demanded).
- -4 if no correct cross edge is present.

Question 3>

- -1.5 mark if not every vertex with in-vertices = 0 is added in the queue
- -1 if you missed decrementing in-vertices
- -1.5 if you missed appending new vertices in the queue

Question 4>

- Full Marks are awarded if the approach is correct and satisfies the required time complexity.
- Partial Marks are awarded if achieved time complexity is incorrect, or it shows a nearly correct approach

Question 5>

- Log (N) ← 2 Marks for Correct Bound (No marks for justification if this is wrong)
- Each vertex will have at least one edge associated with it. So every component will have at least 2 vertices… effectively reducing the no of components by a factor of 2. So No of vertex in graph gets halved in the worst case every time. Therefore log(N)
- No marks for only giving a example in Justification
- You need to explain the reason for log(N)

Question 6>

- No marks based on blanks…. (Marks are awarded according to correctness of algo)
- Decreasing order of Weights (Wi) <= 2 marks for this (No marks for further answer if this is wrong)
- 2 marks for if condition and its outcome.
- 2 marks for else condition
- No marks for just calculating the penalty. (You need to assign the pebble in a slot)
- Expected Answer:

Let L be the pebbles sorted in decreasing order of W_i
Extract the pebble

- If (there exists an empty slot with index <= T_i
  - Place P in rightmost such slot <= T_i
- Else
  - Place P in rightmost slot <= N

Question 7 a>

- -1 mark for frequency values that allow multiple valid combinations in the Huffman coding algorithm, where at least one combination does not produce the desired tree. (–1 for each such case)
- -4 marks if the frequencies defined do not result in the desired tree.

Question 7 b>

Greedy step:

If n is even, replace a_1 and a_2 by a new alphabet a' to get instance A'. f(a')=f(a_1)+f(a_2)

If n is odd, replace a_1, a_2, a_3 by a new alphabet a' to get instance A'.
f(a')=f(a_1)+f(a_2)+f(a_3)

Theorem: If n is even, Opt(A)=Opt(A')+ f(a_1)+f(a_2)

If n is odd, Opt(A)=Opt(A')+ f(a_1)+f(a_2)+f(a_3)

Note:

1. No marks if you replace a_1,a_2,a_3 by a single alphabet.
2. Only partial marks if you carry out two calls : One call with replacing {a_1,a_2}, another call with replacing {a_1,a_2, a_3} and then take the minimum.

   This is because the algorithm may take exponential time.