CS4407

Please answer all questions

Points for each question are indicated by [xx]

1. **[15] Cookie assignment**: Consider the following problem: You are baby-sitting *n* children and have $m > n$ cookies to divide between them. You must give each child exactly one cookie (of course, you cannot give the same cookie to two different children). Each child has a greed factor G_i, $n \geq i \geq 1$ which is the minimum size of a cookie that the child will be content with; and each cookie has a size S_j, $m \geq j \geq 1$. Your goal is to maximize the number of content children, i.e., the number of children such that we have child *i* being assigned a cookie *j* with $S_j \geq G_i$.
   a. **[5]** Define pseudo-code a greedy algorithm to solve the cookie assignment problem.
   b. **[5]** Define the complexity of your algorithm.
   c. **[5]** Is the algorithm optimal? Prove this, or give a counter-example to show sub-optimality.

2. **[15] Oxen pairing**: Consider the following problem: We have *n* oxen, $Ox_1,\ldots,Ox_n$, each with a strength rating S_i. We need to pair the oxen up into teams to pull a plow; if Ox_i and Ox_j are in a team, we must have $S_i + S_j \geq P$, where P is the weight of a plow. Each ox can only be in at most one team. Each team has exactly two oxen. We want to maximize the number of teams.
   a. **[5]** Define pseudo-code for a greedy algorithm to solve the oxen pairing problem.
   b. **[5]** Define the complexity of your algorithm.
   c. **[5]** Is the algorithm optimal? Prove this, or give a counter-example to show sub-optimality. [THIS IS CHALLENGING—TOO HARD FOR AN EXAM]

3. **[15] Pre-emptive scheduling**: Consider the following scheduling problem. You are given *n* jobs. Job *i* is specified by an earliest start time s_i, and a processing time p_i. We consider a pre-emptive version of the problem where a job's execution can be suspended at any time and then completed later. For example if $n = 2$ and the input is $s_1 = 2$, $p_1 = 5$ and $s_2 = 0$, $p_2 = 3$, then a legal pre-emptive schedule is one in which job 2 runs from time 0 to 2 and is then suspended. Then job 1 runs from time 2 to 7 and finally, job 2 is completed from time 7 to 8. The goal is to output a schedule that minimizes

$$\sum_{j=1}^{n} C_j$$

where C_j is the time when job *j* is completed.
In the example schedule given above, $C_1 = 7$ and $C_2 = 8$.
   a. **[5]** Define pseudo-code for a greedy algorithm to solve the optimal pre-emptive scheduling problem.
   b. **[5]** Define the complexity of your algorithm.
   c. **[5]** Is the algorithm optimal? Prove this, or give a counter-example to show sub-optimality. [THIS IS CHALLENGING—TOO HARD FOR AN EXAM]

4. **[15]** Assume that we are given a weighted, directed graph G(V,E), and a non-negative weighting function w: $E \rightarrow \mathbb{Z}^{+}$. We want an algorithm **A** that computes the single-*destination* shortest paths, i.e., the minimum-cost paths **P** from every node to a target node $v \in V$, such that for each path $P_i \in \mathbf{P}$ from node v_i to v, $v_i \neq v$, P_i is of minimum cost among all paths from v_i to v.
    a. **[5]** Define pseudo-code for the required algorithm. You may use Dijkstra's algorithm as a sub-routine, or modify it to your needs.
    b. **[5]** Show that your algorithm is correct.
    c. **[5]** Derive the complexity of the required algorithm **A**, justifying how you obtained your answer.