CSE 450 - Design and Analysis of Algorithms
Quiz 2, Fall 2022
Closed Books, Closed Notes
Time: 45 minutes
Answer any two questions
Each question carries 25 pts.

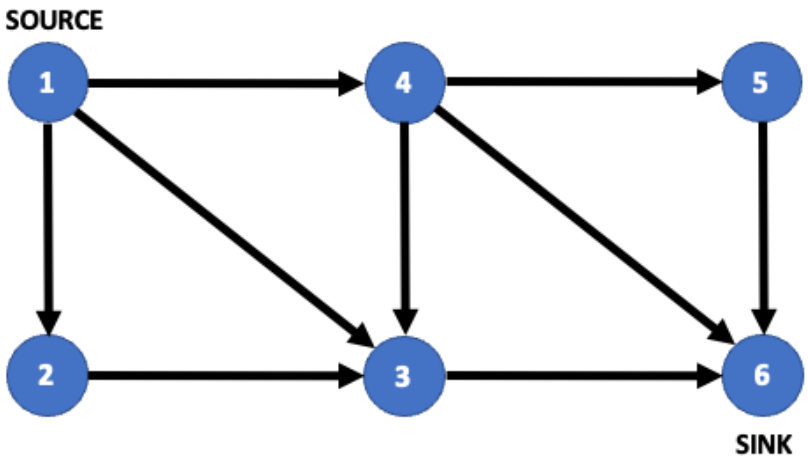
Important Note: *For the purpose of grading your answers, significant emphasis will be given to the process through which you arrive at the answer. In other words, points will be deducted if you do not provide proper justification for your answer, even when you provide the correct answer. If the question asks for finding the answer following a specific technique, points will be deducted if you don't follow that technique but still provide the correct answer.*

Problem 1: Let $G = (V, E)$ be an undirected graph. A node cover of G is a subset U of the vertex set V such that every edge in E is incident to at least one vertex in U. A minimum node cover is one with the fewest number of vertices. Consider the following greedy algorithm for this problem:

procedure *COVER*(V, E)
  $U \leftarrow \emptyset$
  loop
    let $v \in V$ be a vertex of maximum degree
    $U \leftarrow U \cup \{v\}, V \leftarrow V - \{v\}$
    $E \leftarrow E - \{(u, w)$ such that $u = v$ or $w = v\}$
  until $E = \emptyset$ **repeat**
  return(U)
  end *COVER*

Does this algorithm always generate a minimum node cover? If your answer is "Yes" then prove it by providing arguments as to why this algorithm will always find the minimum node cover. If your answer is "No" then provide an example where the algorithm fails to find the minimum node cover.

Problem 2:



Consider the network given in the figure above. A *directed edge* in a directed graph (as shown above) is also know as an *Arc*. Assume that the *capacity* each arc in the graph above is 1.

(a) Two directed paths P_1 and P_2 from a source node s to a sink node t is called *arc-disjoint* if the paths do not have any common arc. Using Max-Flow algorithm, compute the maximum number of arc-disjoint paths from the source node to the sink node in the above graph. Show all your work (i.e., generation of the augmenting paths to increase flow from the source to the sink node).

(b) Enumerate all $s-t$ cuts in the network. For each $s-t$ cut $[S, \bar{S}]$, list the nodes and the sets of forward and backward arcs. (Forward arcs are the ones whose edge orientation is from a node in S to a node in $\bar{S}$, and Backward arcs are the opposite.)

(c) Verify that the maximum number of arc-disjoint paths from the source node s to the sink node t equals the minimum number of forward arcs in an $s-t$ cut.

Problem 3: Given the job scheduling problem with the following specifications: a set of jobs $J = J_1, ..., J_n$, their respective execution times $t_1, ..., t_n$, deadlines $d_1, ..., d_n$ and profits $p_1, ..., p_n$. The profit associated with a job is realized only if a schedule can finish jobs before their respective deadlines. If a job finishes after its deadline, no profit is made. The goal here is to maximize the profit.

- Consider the situation where $t_1 = t_2 = ... = t_n$.
- Consider the situation where $t_1 \neq t_2 \neq ... \neq t_n$.

Develop algorithms for both situations. Analyze your algorithms to establish, (i) *correctness* (i.e., it does what it's supposed to do, i.e., to select a (sub)set of J that will maximize profit for all problem instances), and (ii) *computational complexity* (i.e., amount of computation that will be needed to find the correct solution).