Chennai Mathematical Institute

Design and Analysis of Algorithms CU2101

End-Term

Aug-Nov 2025 Time: 2:00 HRs Max mark: 40

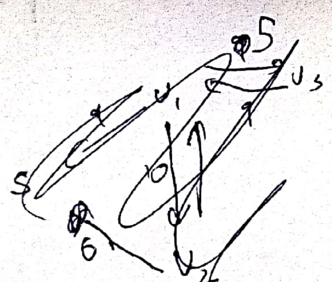
Name:

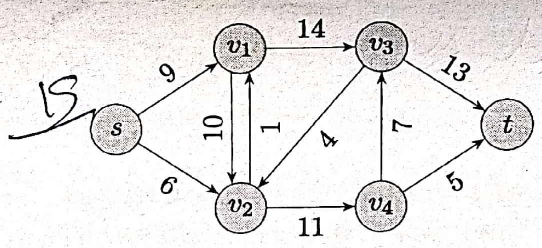
Roll Number:

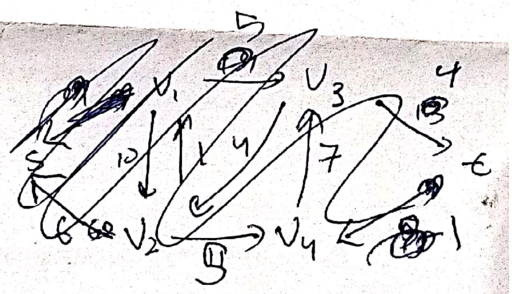
- **Answer all questions.**
- **Your answers for subjective questions should be clear, precise and brief.**
- **For algorithmic questions, pseudo-code is not required, a concise pseudo-code styled step by step description in English is sufficient.**
- **Always analyse the run-time and prove correctness of the algorithm.**

1. Let $G = (V, E, c)$ be a flow network, s and t be the source and the sink, $f : V \times V \rightarrow \mathbf{R}$ be a flow, c is the capacity, $|f|$ is the value of flow f and (S, T) be a cut. Answer the following. (7)

   a) What is the maximum flow in the following network? Write all the edges and the corresponding flows of your solution. (3)

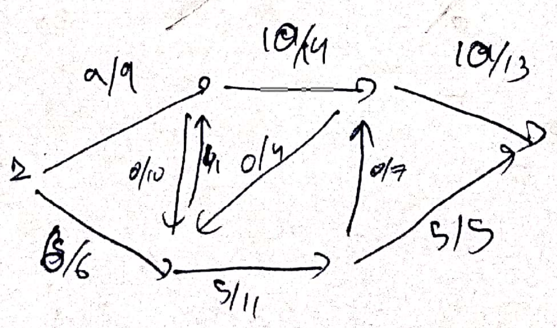






   b) Prove that if the residual graph G_f has no augmenting path then $|f| = c(S, T)$ for some cut in G. (4)

2. Given a weighted directed graph $G = (V, E, w)$, a *longest simple path* p from u to v is a path such that total sum of the weights of edges in p is maximum and there are no repeated vertices (i.e. no cycles) in p. (8)



a) Give a linear time ($O(|V|+|E|)$) algorithm to compute the single source longest path in a directed acyclic graph. (4)

b) Can the longest simple path problem be solved in a general weighted directed graph in polynomial time? Justify your answer. (4) red from HP

3. A *bottleneck spanning tree* T of a weighted undirected graph G is a spanning tree of G whose largest edge weight is minimum over all spanning trees of G. The **value** of the bottleneck spanning tree is defined as the weight of the maximum-weight edge in T. (10)

   a) Prove that a minimum spanning tree is a bottleneck spanning tree. (2) easy

   b) Give an $O(|V|+|E|)$-time algorithm that given an integer b, determines whether the value of the bottleneck spanning tree in G is at most b. (4) Delete then BFS

   c) Use your algorithm for part (b) as a subroutine in a linear-time algorithm for the bottleneck-spanning-tree problem. (4)

4. A spanning tree T of a connected graph G rooted at v is called a *depth-first spanning tree* (DFS tree) if it arises from exploring vertices using depth-first search; it is called a *breadth-first spanning tree* (BFS tree) if it arises from exploring vertices using breadth-first search.

   Let G be a connected graph, and let T be a depth-first spanning tree of G rooted at some node v. Prove that if T is also a breadth-first spanning tree of G rooted at v, then $G = T$. induction + BFS(Δ) ≠ DFS(Δ) (7)

5. Consider an expression consisting of integers separated by + and − signs, for example $1 + 3 - 2 - 5 + 1 - 6 + 7$. By inserting parentheses in different positions, the value of the expression may change. For instance:

   $$(1 + 3 - (2 - 5)) + (1 - 6) + 7 = 9, \qquad (1 + (3 - 2)) - (5 + 1) - (6 + 7) = -17.$$

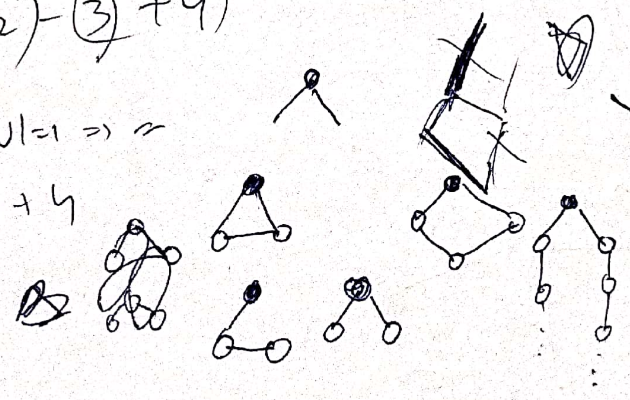
   Parentheses may only be used to group additions and subtractions; in particular, they may not be used to create implicit multiplication such as $1 + 3(-2)(-5) + 1 - 6 + 7$.

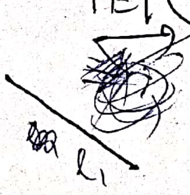
   Describe and analyze an efficient polynomial-time algorithm (in terms of the number of integers in the input expression) which, given such an expression, computes the maximum possible value obtainable by inserting parentheses. (8)

((1+2)−3+4)

|V|=1

(1+2)−3+4

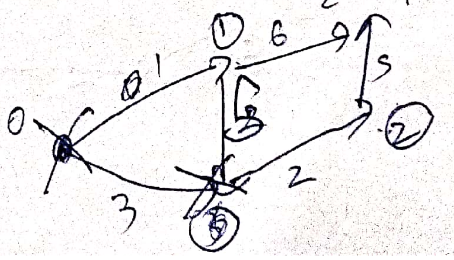




$|E|\cdot(|V|+|E|)$

$e_2 < l_2$

$l_1 < l_1$



(1−(1+2))