CSE 421 Spring 2025: Set 6

Instructor: Chinmay Nirkhe
Due date: May 21st, 2025 11:59pm

Instructions: Solutions should be legibly handwritten or typeset (ideally in LaTeX). Mathematically rigorous solutions are expected for all problems unless explicitly stated.

You are encouraged to collaborate on problems in small teams but everyone must individually submit solutions. Solutions for the problems may be found online or in textbooks – but do not use them.

For grading purposes, list, with each problem, the names of your collaborators. Please start each problem on a new page.

Problem 1 (Minimum tutoring sessions)**.** Consider a school with n_1 students $a_1, \dots, a_{n_1}$ and n_2 tutors $b_1, \dots, b_{n_2}$. Each tutor b_j is qualified to supervise a subset of the students, expressed as a list x_j. Each student needs only one tutor and there exists some tutor qualified to teach each student.

1. **[10 points]** Give an algorithm for computing the maximal number of students that can be paired with a qualified tutor if each tutor can supervise at most k students. Give the runtime and proof of correctness. The runtime should be expressed in terms of n_1, n_2, k and m = the number of valid student-tutor pairs.

2. **5 points]** Use the algorithm from the first part to compute the minimal value k such that every student can be guaranteed a tutor. Give the runtime and proof of correctness (this should be short given the previous part).

Problem 2. [10 points] Suppose you are given an arbitrary flow network on a network (G, c, s, t) where the capacities of c are integers. Complete the following sentences and justify your answers:

1. *Decreasing the capacity of an edge by 1 decreases the size of the max flow if and only if the edge belongs to a ...*
2. *Increasing the capacity of an edge by 1 increases the size of the max flow if and only if the edge belongs to ...*

Note that this problem is in terms of a flow network but no specific max flow $f_{\max}$ has been described.

Problem 3 (Minimum disconnecting cut). **[10 points]** Given an undirected graph $G = (V, E)$, construct an algorithm for computing the minimum number of edges that need to be removed in order to disconnect the graph. Your algorithm should run in time $O(mn^2)$. Prove the runtime and correctness.

Problem 4 (Exit strategies)**.** An ant colony is a complex maze best represented by a directed unweighted graph $G = (V, E)$ with n vertices and m edges. Assume that, initially, there is exactly one ant at each vertex. Suddenly an anteater emerges and all the ants need to escape. Having planned for this, they have built some special panic rooms $P \subsetneq V$ where the anteater can't reach them. Assume that each panic room $v \in P$ only holds k ants.

1. **[5 points]** Construct an algorithm for calculating the maximum number of ants that could successfully escape to a panic room. Give the algorithm description and runtime in terms of $n, m, |P|$ and k but ignore the proof of correctness.

2. **[10 points]** Let us make the problem more realistic: Assume that if the ants don't escape to a panic room in time $T \in \mathbb{N}$, they will be eaten. Also, an ant can move from vertex to vertex in unit time or an ant can choose to stay at it's vertex.

   Give an algorithm for calculating the maximum number of ants that could successfully escape to a panic room. Give the algorithm description, runtime, and proof of correctness. This time the runtime will depend on (n, m, k, T).

   Hint: Create a new graph with $O(Tn)$ vertices and run a flow algorithm on the new graph.

3. **[3 points]** Next, what is a modification you can make to your previous solution such that, at any moment in time, at most 10 ants can be at any vertex $v \notin P$? You can also assume there are no edges leaving panic rooms. State the modification but do not prove runtime or correctness. [2-3 sentences suffice.]

General hints: Draw a picture. Personally, I like to think of this problem as a tall building with the ants initially all starting at the top floor and have to get to the panic rooms at the bottom floor. Perhaps this will help you too.

Problem 5. [5 extra credit points] Given a flow network (G, c, s, t) with $c : E \to \mathbb{Z}_{\geq 0}$, let us introduce a second cost function $\$: E \to \mathbb{R}$. Therefore, each edge e has a capacity $c(e)$ as well as a cost $\$(e)$. Note, the cost of an edge may be negative.

There may be multiple maximum flows in the network. For each flow f, let $\$(f)$ equal $\sum_{e \in E} f(e)\$(e)$. Construct an algorithm for computing the minimum cost flow amongst the set of maximum flows.

Hint: Start with any integer maximum flow f. Then lower the cost $\$(f)$ of the flow along a directed cycle containing s and t using a variant of the Bellman-Ford algorithm on the residual graph G_f.