CSE 421 Spring 2025: Set 5

Instructor: Chinmay Nirkhe
Due date: May 14th, 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 (Finding a negative cycle)**.** In class, I showed how to detect a negative cycle in a directed graph with negative weights. Chapter 6.10 of Kleinberg and Tardos shows how to convert this algorithm into one for finding a negative cycle. Spend some time with Kleinberg and Tardos until you think you understand the algorithm and its proof of correctness.

[10 points] Then, in your own words, write out the algorithm and proof of correctness. We don't need to see your runtime analysis.

Problem 2 (Target hitting)**.** In this problem, we have n items with positive integer values $v_1, \dots, v_n$ that are all different. There is also a positive integer target value V. The goal in this question is to figure out the *number of different subsets* $S \subseteq [n]$ such that $\sum_{i \in S} v_i = V$.

1. **[10 points]** Give an algorithm with running time $O(nV)$ to produce this answer. Prove runtime and correctness but don't worry about the space complexity.
2. **[5 points]** How would you modify your solution to keep the same runtime and use space $O(V + n)$? No need to prove correctness; just state the algorithmic modifications.

Problem 3. The *Cat and Mouse* game is played on an undirected connected graph G. The Cat and the Mouse each initially occupy designated vertices and take turns moving along edges of the graph. The Cat wins if at any point it is on the same vertex as the Mouse. The Mouse wins if it avoids the Cat forever. Each animal must move during its turn.

Suppose the graph has n vertices and $m \geq n$ edges. We say that a starting configuration (G, v, w, b) (in which G is the graph, v is the Cat position, w is the Mouse position, and b is a bit which is 0 if its the Cat's turn to move, and 1 if it's the Mouse's turn to move) is a *Cat-win configuration* if the Cat can *force* a win. That is, it is a Cat-win configuration if, no matter what moves the Mouse makes, the Cat will be able to Catch him. If a configuration is not a Cat-win, then we call it a Mouse-win configuration.

1. **[15 points]** Describe an algorithm (with an asymptotic runtime mn^3 or better), to determine which player wins — i.e., the input is a starting configuration (G, v, w, b) and you need to determine whether this is a Cat-win or Mouse-win configuration. Prove runtime and correctness. You do not need to optimize for space complexity.

   Hint: If you are stuck, try the following systematic approach.

   (a) Find an upper bound on the number of steps it takes for the Cat to catch the Mouse (if it ever does).

   (b) Consider a subproblem described by the locations of the Cat and Mouse, whose turn it is, and a bound on how many steps the Cat has to catch the Mouse.

   (c) Build a dynamic programming algorithm based on the bases case of the animals being at the same location.

2. **[2 extra credit points]** We award two extra credit points if you can come up with an algorithm with an asymptotic runtime of mn or better.