

Final Exam

Instructors: Shuchi Chawla & Christos Tzamos

Due: May 7, 2020, 11:59 p.m.

Guidelines:

- You have five days to complete this exam. Please upload your solutions in PDF format on Canvas by the due date.
- The exam is due by May 7, 2020, 11:59 p.m. No extensions will be provided under any circumstances.
- You are required to answer each of the THREE multi-part questions.
- You can use all the results we showed in class or in the homework. Clearly state the results you use. Substantiate all other claims you make.
- You are not allowed to consult any material outside of material provided for this course. Any evidence of cheating or plagiarism will be dealt with utmost seriousness.
- For your convenience, at the back of this exam we have included a list of decision problems proven to be NP-Complete in class.

GOOD LUCK!

Problems:

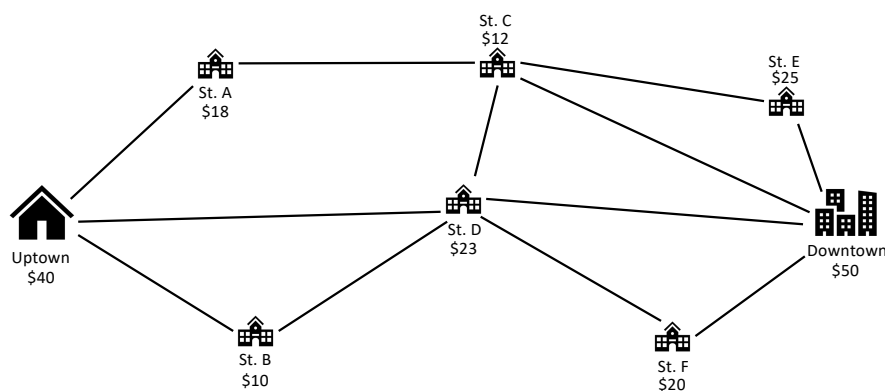
1. **[3+1+1+1 points]** You have designed a new single-player multi-level game of strategy and chance for your gaming startup where players aim to maximize the number of points they collect before completing all levels. Now you would like to design an algorithm to compute the most number of points a player can obtain in the game in expectation by playing an optimal strategy.

Here is how the game works. There are n levels. In each level i , the player has m actions available. The j th action awards the player P_{ij} points but takes the player to a random level above i . In particular the player has a probability π_{ijk} of ending up at a level $k > i$ upon following action j in level i . The game ends when the player reaches level n . A strategy for the player assigns an action to follow at every level i .

In this question you will develop a dynamic program for solving this problem. For full credit, your algorithm should run in time $\text{poly}(n, m)$.
 - (a) Define a function that computes the expected total points obtained by the optimal player strategy. Provide recursive equations for computing the function.
 - (b) Briefly explain (in 1-2 sentences) why your recursive equations are correct.
 - (c) Write a dynamic program based on your recursive equations to compute the expected total points obtained by the optimal player strategy.
 - (d) State the running time of your program.
2. **[3+1+1 points]** Residents of the city of TrainVille use its extensive public transportation system to commute from home to work everyday. Stations throughout the city are interconnected by subway lines. An advertising company in the city of TrainVille wants to target its ads to all commuters going from the Uptown station to the Downtown station. There are multiple routes going from Uptown to Downtown, and different stations charge different amounts for displaying ads. The advertising company's goal is to ensure that along *any* such route, at least one station carries their ad, while minimizing the total amount of money they spend.

In this question, you will develop a polynomial time algorithm for this problem by reducing it to the network flow problem. Your algorithm is given as input a subway map for the city of TrainVille that shows all of the train connections between the n stations in the city, as well as the price of advertisement at each individual station.

- Describe a reduction from the advertising problem to max s - t flow or min s - t cut. Remember to specify how to convert a solution of your reduced instance into a solution to the advertising problem.
- What does your network flow instance look like when given the map below? What is the optimal solution in this map?
- Give a brief (1 paragraph) argument for the correctness of your reduction.



- [1+2+2 points]** You go into a store and want to buy several items. The store has n different items. Item i is priced at $\$p_i$, and you have a value of $\$v_i$ for this item. You want to purchase a set S of items so as to maximize your net utility from the purchase—the total value you get minus the price you pay. In order to do so, you should buy precisely the items for which $v_i > p_i$. However, there is a catch. If you spend a total of at least T dollars, you get a 10% discount on your total payment. So it may make sense to buy some items for which $v_i \leq p_i$.

For example suppose that the store has three items, each of which you value at \$10. Suppose that the three items are priced at \$9, \$11, and \$13 respectively, and the discount threshold T is \$15. Then, without the discount you would want to buy only the first item, which brings you a net utility of \$1. However, with the discount, it makes sense to buy the first two items. This brings you a total value of \$20 at a price of $90\% \times \$20$, or a net utility of \$2.

In the decision version of this problem, you are given the n positive integral prices, n positive integral values, the discount threshold T , and a utility target U . Your goal is to determine whether there is a set S of items that brings you utility at least U . You will prove that this problem is NP-hard by providing a mapping reduction.

- Specify which problem you will reduce to what.
- Describe your mapping reduction.
- Prove that the mapping reduction is correct. (Remember that you need to show two implications.)

For your reference, here is a (partial) list of NP-Complete problems discussed in class and their definitions.

- **3-SAT:** Given a 3-CNF formula, is the formula satisfiable?
- **Vertex Cover:** Given a graph G and target k , does the graph contain a vertex cover of size at most k ? A vertex cover is a subset of vertices that includes at least one end point of every edge in the graph.
- **Independent Set:** Given a graph G and target k , does the graph contain an independent set of size at least k ? An independent set is a subset of vertices such that no two vertices in the subset are connected by an edge.
- **Clique:** Given a graph G and target k , does the graph contain a clique of size at least k ? A clique is a subset of vertices such that every pair of vertices in the subset is connected by an edge.
- **Subset Sum:** Given n positive integers x_1, \dots, x_n and a target k , is there a subset $S \subseteq [n]$ such that the numbers corresponding to the indices in S add up exactly to k , that is, $\sum_{i \in S} x_i = k$?
- **Knapsack:** Given n positive integer values v_1, \dots, v_n ; n positive integer weights w_1, \dots, w_n ; knapsack capacity C ; and a target k , is there a subset $S \subseteq [n]$ such that $\sum_{i \in S} w_i \leq C$ and $\sum_{i \in S} v_i \geq k$?