Assignment III for AI2615 (Spring 2022)

April 4, 2022

Due: Monday, Apr. 25, 2022.

Problem 1 (20 points)

A server has n customer waiting to be served. The service time required by each customer is known in advance: it is t_i minutes for customer i. So if, for example, the customers are served in order of increasing i, then the ith customer has to wait $\sum_{j=1}^{i} t_j$ minutes.

We wish to minimize the total waiting time

$$T = \sum_{i=1}^{n} (\text{time spent waiting by customer } i.)$$

Give an efficient algorithm for computing the optimal order in which to process the customers. Prove the correctness of your algorithm.

Problem 2 (35 points)

In the class, we learnt the Kruskal's algorithm to find a minimum spanning tree (MST). The strategy is simple and intuitive: pick the best legal edge in each step. The philosophy here is that local optimal choices will yield a global optimal. In this problem, we will try to understand to what extent this simple strategy works. To this end, we study a more abstract algorithmic problem of which MST is a special case.

Consider a pair $M = (U, \mathcal{I})$ where U is a finite set and $\mathcal{I} \subseteq 2^U$ is a collection of subsets of U. We say M is a *matroid* if it satisfies

- (**hereditary property**) $\mathcal{I}$ is nonempty and for every $A \in \mathcal{I}$ and $B \subseteq A$, it holds that $B \in \mathcal{I}$.
- (**exchange property**) For any $A, B \in \mathcal{I}$ with $|A| < |B|$, there exists some $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$.

Each set $A \in \mathcal{I}$ is called an *independent set*.

A set $A \in \mathcal{I}$ is called *maximal* if there is *no* $B \in \mathcal{I}$ such that $A \subsetneq B$.

1. (5 points) Let $M = (U, \mathcal{I})$ be a matroid. Prove that maximal independent sets are of the same size.

Hint: When proving the *exchange property* of two independent sets A, B, consider the forest induced by A in G.

2. (5 points) Let $G = (V, E)$ be a simple undirected graph. Let $M = (E, \mathcal{S})$ where $\mathcal{S} = \{F \subseteq E \mid F \text{ is acyclic}\}$. Prove that M is a matroid. What are maximal sets of this matroid?

3. (10 points) Let $M = (U, \mathcal{I})$ be a matroid. We associate each element $x \in U$ with a non-negative weight $w(x)$. For every set of elements $S \subseteq U$, the weight of S is defined as $w(S) \triangleq \sum_{x \in S} w(x)$. Now we want to find a maximal independent set with maximum weight. Consider the following greedy algorithm.

```
Algorithm: Find a maximal ind. set with maximum weight
Input  : A matroid M = (U, I) and a weight function
         w : U → R≥0.
Output : A maximal ind. set S ∈ I with maximum w(S).

S ← ∅;
Sort U into decreasing order by weight w;
for x ∈ U in decreasing order of w do
  | if S ∪ {x} ∈ I then S ← S ∪ {x};
end
return S;
```

Now we consider the first element x the algorithm added to S. Prove that there must be a maximal independent set $S' \in \mathcal{I}$ with maximum weight containing x.

Hint: Prove by contradiction and use the exchange property.

4. (5 points) Use above algorithm to solve the MST problem and convince yourself that it is equivalent to Kruskal's algorithm.

5. (10 points) Let $U \subseteq \mathbb{R}^n$ be a finite collection of n-dimensional vectors. Assume $m = |U|$ and we associate each vector $\mathbf{x} \in U$ a positive weight $w(\mathbf{x})$. For any set of vectors $S \subseteq U$, the weight of S is defined as $w(S) \triangleq \sum_{\mathbf{x} \in S} w(\mathbf{x})$. Design an efficient algorithm to find a set of vectors $S \subseteq U$ with maximum weight and all vectors in S are linearly independent.

Problem 3 (25 points)

Suppose you are a driver, and you plan to drive from A to B through a highway with distance D. Since your car's tank capacity C is limited, you need to refuel your car at the gas station on the way. We are given n gas stations on the highway with surplus supply. Let $d_i \in (0, D)$ be the distance between the starting point A and the i-th gas station. Let p_i be the price for each unit of gas at the i-th gas station. Suppose each unit of gas exactly supports one unit of distance. The car's tank is empty at the beginning, and the 1-st gas station is at A. Design efficient algorithms for the following tasks.

1. (10 Points) Determine whether it is possible to reach B from A.

2. (15 Points) Minimized the gas cost for reaching B.

Please prove the correctness of your algorithms and analyze their running times. You are asked to implement your algorithm as efficient as possible.

Problem 4 (20 points)

Given a constant $k \in \mathbb{Z}^+$, we say that a vertex u in an undirected graph *covers* a vertex v if the distance between u and v is at most k. In particular, a vertex u covers all those vertices that are within distance k from u, including u itself. Given an undirected *tree* $G = (V, E)$ and the parameter k, consider the problem of finding a minimum-size subset of vertices that covers all the vertices in G.

1. (10 points) Design an efficient algorithm for the problem above with $k = 1$.
2. (10 points) Design an efficient algorithm for the problem above with general k.

Please prove the correctness of your algorithms and analyze their running times.

Problem 5

How long does it take you to finish the assignment (including thinking and discussion)?

Give a rating (1,2,3,4,5) to the difficulty (the higher the more difficult) for each problem.

Do you have any collaborators? Please write down their names here.