CS 170<br>Spring 2017 | Efficient Algorithms and Intractable Problems<br>Prasad Raghavendra and Sanjam Garg | Final Review

May 6, 2017

## 1. DAG Revisited

Design an algorithm that takes a directed acyclic graph, $G = (V, E)$, and determines whether $G$ contains a directed path that touches every vertex exactly once. The algorithm should run in $O(|V| + |E|)$ time.

## 2. Another MST Algorithm

In this problem, we will develop a new algorithm for finding minimum spanning trees. It is based upon the following property:

Pick any cycle in the graph, and let $e$ be the heaviest edge in that cycle. Then there is a minimum spanning tree that does not contain $e$.

(a) Prove this property carefully.

(b) Here is the new MST algorithm. The input is an undirected graph $G = (V, E)$ with edge weights $\{w_e\}$.

**In decreasing order of edge weights, if edge e is part of a cycle, remove it from G. When you cannot remove any more edges, return G.**

Prove that this algorithm is correct.

(c) What is the overall time taken by this algorithm, in terms of $|E|$?

## 3. Minimal Graphs

(a) A city has $n$ intersections, and $m$ undirected roads. However, road maintenence is getting really expensive, so the city would like to reduce the number of roads. They want to do this without affecting the overall connectivity of the intersections.

More formally, you are given as input a graph $G = (V, E)$, with edges undirected. You would like to create a new graph $G' = (V, E')$, with edges undirected and $|E'|$ minimized with the following constraint that if there exists a path from $u$ to $v$ in $G$, then there exists a path from $u$ to $v$ in $G'$.

Give an efficient algorithm to determine the minimum number of undirected edges we need in the modified graph. Note that if there isn't a path from $u$ to $v$ in $G$, then there may or may not be a path from $u$ to $v$ in $G'$.

(b) Repeat the same excercise, but now for directed graphs. More specifically, you are given as input a graph $G = (V,E)$ with edges **directed**. You would like to create a new graph $G' = (V,E')$ with edges **directed** and $|E'|$ minimized with the following constraint that if there is a path from $u$ to $v$ in $G$, then there must be a path from $u$ to $v$ in $G'$. (Note that we don't necessarily need a path from $v$ to $u$).

Give an efficient algorithm to determine the minimum number of directed edges we need in the modified graph. Note again that if there isn't a path from $u$ to $v$ in $G$, then there may or may not be a path from $u$ to $v$ in $G'$.

## 4. Revisiting zero-sum games

Suppose we have a zero-sum game with payoff matrix $M$. The row player tries to maximize the expected payoff whereas the column player tries to minimize it. Under the following conditions, indicate (YES or NO) whether entry $m_{i,j]}$ is the optimal expected payoff value in general:

| Condition | Is $m_{i,j}$ optimal? |
|---|---|
| $m_{i,j}$ is **smallest** in row $i$ and **smallest** in column $j$ | |
| $m_{i,j}$ is **largest** in row $i$ and **smallest** in column $j$ | |
| $m_{i,j}$ is **smallest** in row $i$ and **largest** in column $j$ | |
| $m_{i,j}$ is **largest** in row $i$ and **largest** in column $j$ | |

## 5. Assigning workers

Assume we have $N$ workers. Each worker is assigned to work at one of $M$ factories. For each of the $M$ factories, they will produce a different profit depending on how many workers are assigned to that factory. We will denote the profits of factory $i$ with $j$ workers by $P_i^j$.

(a) How would you find the assignment of workers that produces the most profit?

(b) How can this algorithm be improved if we enforce the law of diminishing returns? Namely that for each factory each additional worker (past the first worker) cannot result in a larger increase in profits than the previous worker. (For example, $P_1^1 = 5$ and $P_1^2 = 7$ obeys diminishing returns as the first worker adds profit 5 and the second adds profit 2.)

6. **Assigning backups**

Horizon Wireless is building a 5G network. The company has a set $V$ of wireless towers; the distance $d(i,j)$ between any two of them is known, and each tower is capable of transmitting to other towers within a distance $r$.

- To make this network fault-tolerant, Horizon wants to assign each tower $v \in V$ to **two** other backup towers, so that if $v$ is about to fail, it can transmit its data to them.
- Due to storage constraints, each tower can only serve as a backup for up to **three** other towers.

(a) Suppose Horizon partitions its towers $V$ into active towers $A$ and backup towers $B$. Devise an algorithm which, given $V = A \cup B$, a distance function $d(i,j)$ on $V$, and tower radius $r$, assigns each active tower $a \in A$ to two backup towers in $B$ (subject to the storage constraint), or reports that no assignment is possible. (*Hint: build a graph and use a known algorithm.*

(b) To use its network more efficiently, Horizon wants to use all towers in $V$ as active towers, but still wants to assign each tower $v \in V$ to two other towers in $V$ as backups. Again, no tower can be a backup for more than three other towers. Give an algorithm to find the assignment of backups for every tower.