**CSC 373 - Algorithm Design, Analysis, and Complexity** **Summer 2016**

# Assignment 2: Due Tuesday July 5, 10PM

**Please follow the instructions provided on the course website to submit your assignment**. You may submit the assignments in pairs. Also, if you use **any** sources (textbooks, online notes, friends) please cite them for your own safety.
You can use those data-structures and algorithms discussed in CSC263 (e.g. merge-sort, heaps, etc.) and in the lectures by stating their name. You do not need to provide any explanation or pseudo-code for their implementation. You can also use their running time without proving them: for example, if you are using the merge-sort in your algorithm you can simply state that merge-sort's running time is $\mathcal{O}(n \log n)$.

Every time you are asked to design an efficient algorithm, you should provide both a short high level explanation of how your algorithm works in plain English, and the pseudo-code of your algorithm similar to what we've seen in class. State the running time of your algorithm with a brief argument supporting your claim. You must prove that your algorithm finds an optimal solution!

## 1 Fight Fight Fight ... Or come up with an algorithm.

You're at a fair with a group of friends, when you saw a booth selling a booklet of tickets to a Eurocup match. Excited, you decided to check it out. Unfortunately with all your coins combined, you didn't have enough to buy the booklet. The vendor challenged you[1] to play a game and if you win, you win the booklet. You won!

Once the booklet in hand, you realized it didn't have enough tickets for all of you to go. Luckily [2], there is an even number of you $n$, and the booklet has $n/2$ tickets. You decided the fairest way to get tickets is to split into two groups, and play the following game, and the team to win the most money gets the booklet.

The game goes as follows: You each put a coin down. The coins are placed in a row. Your team goes first. At every step of the game you can take a coin from an end point of the row. The other team goes next and does the same. You keep alternating until no coins are left. The ordering of the coins is arbitrary, and the coins don't necessarily have equal value. Give an algorithm that allows your team to win as much as possible. Prove it is correct.

## 2 Thank you for your help.. Choos.

After a large scale fire that destroyed your city, the injured residents were taken to hospitals at nearby cities. You joined a volunteer organization that decided to send a group of $n$ volunteers, including yourself, to help at these hospitals. Since this is taking place outside the city, the organizers decided to assign volunteers to hospitals within a half-hour drive from their home. The hospitals in the other hand decided to provide free meals to all the volunteers, but due to budget constraints, each hospital cannot feed more than $\lceil \frac{n}{m} \rceil$ volunteers, where $m$ is the total number of hospitals involved in this rescuing operation.

Give an algorithm that allows the organizers to decide whether it is possible to satisfy all these constraints. Analyze its running time in term of $n$ and $m$.

[1] it's a fair after all
[2] or not

# 3 More Graphs ❤

Given a graph $G(V, E)$, a vertex cover is a subset $S \subseteq V$ of vertices such that every edge $e \in E$ has an endpoint in $S$. We say that $e$ is covered, hence the name vertex cover.

A $k$-regular graph is a graph where every vertex $v$ has exactly $k$ neighbours; i.e. $\forall v, |\{u|(u, v) \in E\}| = k$. We often drop the $k$, and call $G$ regular.

A perfect matching is a matching where every vertex is covered by an edge.

1/ Prove that if $G(A \cup B, E)$ is bipartite, the size of the maximum matching equals the size of a minimum vertex cover.

2/ Let $G(A \cup B, E)$ be a bipartite graph. For a set $S \subseteq A$, let the set $N(S)$ be the set of neighbours of $S$, i.e.

$$N(S) = \{v \in V | \exists u \in S, (u, v) \in E\}$$

Show that $G$ has a matching in which every vertex of $A$ is matched iff for every subset $S$ of $A$:

$$|N(S)| \geq |S|$$

3/ Show that any non-trivial regular bipartite graph has a perfect matching.

# 4 Cha Cha Cha Chaaanges

Let $(G, s, t, c)$ be a flow network with integral capacities, and let $f$ be its maximum flow.

1/ Suppose we increase a specific capacity by 1. Show how to compute the new maximum flow in $\mathcal{O}(m)$ time, where $m$ is the set of edges in $G$.

2/Suppose now we decreased a capacity ($\geq 1$) by 1. Show how to compute the new maximum flow in $\mathcal{O}(m)$ time.

# 5 What Goes Up, Must Come Down... Or does it?

Let $(G, s, t, c)$ be a flow network with vertex set $V$, and edge set $E$.

1/ Let $E^+ \subseteq E$ denote the subset of edges where for all $e \in E^+$, increasing $c(e)$ increases the maximum flow of $G$. Is $|E^+| > 0$ for all $G$? Give an algorithm that finds all $e \in E^+$, with a running time better than $m$ max-flows.

2/ Similarly, let $E^- \subseteq E$ denote the subset of edges where for all $e \in E^-$, decreasing $c(e)$ decreases the maximum flow of $G$. Is $E^- = E^+$? Give an algorithm that finds all $e \in E^-$, and analyze it.