CPSC 320 2016W2: Quiz 3 Pre-Release Information

February 7, 2017

This week's quizzes will follow up on the problem domains described below. It's worth spending a few minutes reading and understanding each domain before your tutorial!

Note that if you collaborate on understanding this information, you should follow the academic conduct guidelines. Among other rules, take down the GradeScope student #s or account names of collaborators for acknowledgement, but take **no other notes** away from those collaborators!

NOTE: The names of problems (on the quiz) subproblems are often intended to be fun and are completely irrelevant. Feel free to ignore them here and on the quiz!

Please, remember your GradeScope # when coming to the tutorial quiz.

1 Greedy banks resequencing debits

Predatory banks take the debits to an account that occur over the day and reorder them to maximize the fees they can charge. For each debit that results in taking an account into overdraft (having negative balance in the account) or where the account is already in overdraft, the bank charges the customer an overdraft fee.

2 Parking optimization in Wonderland

A company called Wonderland offers several types of parking permits to its employees, with different durations and prices. The coop student Alice will work in Wonderland for n consecutive days. She wants to figure out the cheapest collection of parking permits that would cover all days she needs to be present at work. Alice can buy as many permits of a given type as she likes.

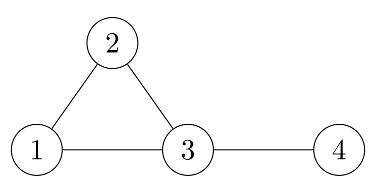
Let's assume there are k type of permits $1, \ldots, k$: the price of permit type t is p_t dollars and duration D_t. Alice needs to stay at work on n consecutive days $T = [1, 2, 3, \ldots, n-1, n]$. ($T$ is for "time period".) Our goal is to help Alice to compute a collection of permits to buy and when to activate them of the form (t, d), where t is the permit type and d is the starting day for that permit. Of course, this collection of permits has to cover all days in T.

3 Coloring graphs with lots of colors

We are given a graph $G = (V, E)$. We want to color each vertex with one of k colors so that two end points of any edge in E receive different colors. This is called a *vertex k-coloring* of the graph.

Recall that the *degree* of a vertex $v \in V$ is the number of edges incident on v. Let Δ be the *maximum degree* in the graph.

Example:



The maximum degree Δ in this graph is 3 and the graph has a 3-coloring and a 4-coloring, but does not have a 2-coloring.

4 Compare running times of the algorithms on graphs

Consider the following pairs of functions representing runtimes of algorithms that take a graph with n vertices and m edges as the input (simple graph though, no self-loops, no directed edges, no multiple edges between the same pair of vertices). For each pair, write between them the best choice of:

- **LEFT** to indicate that the left one is big-O of the right one, i.e., left $\in$ O(right)
- **RIGHT** to indicate that the right one is big-O of the left one, i.e., right $\in$ O(left)
- **SAME** to indicate that two are Θ of each other, i.e., left $\in \Theta$(right)

- **INCOMPARABLE** to indicate that none of the previous relationships holds for all possible values of n and m.

Do not write **LEFT** or **RIGHT** if **SAME** is true.

Examples:

For **any** graph we have:

Left function	LEFT/RIGHT/SAME/INCOMPARABLE	Right function
$n+m$	RIGHT	m

For a graph that is **a tree** we have:

Left function	LEFT/RIGHT/SAME/INCOMPARABLE	Right function
$n+m$	SAME	m

5 Construction site workforce optimization

A construction company is trying to solve the following cost optimization problem. The construction will take place over the next k days. Different number of workers required each day at the construction site: $n_1, \ldots, n_k$. Hiring a worker cost $h > 0$ dollars, discharging a worker costs $d > 0$ dollars. Each worker gets paid $s > 0$ dollars per day, whether it's idle or not. However, if the number of hired workers is less than the required number for that day, some of the hired workers must work overtime and they get paid double rate ($2s$ dollars) for the overtime. Each hired worker can take at most 2 shifts per day (one normal shift and one overtime shift). Hence, the number of hired workers on day i cannot drop below $n_i/2$. The construction company wants to determine the optimal number of workers for each day that would minimize the total cost of the workforce, including daily salary and the cost of hiring and discharging workers. On day 0, no workers are hired.

The *Shortest paths from a single source in a graph problem* takes as the input a directed graph $G = (V, E)$, a weight function $w : E \rightarrow \mathbf{R}^+$ that assigns to every edge a positive real number (weight) and a source vertex s, and returns a shortest path from s to every other vertex in the graph. This problem can be solved efficiently by Dijkstra algorithm.