CME 305: Discrete Mathematics and Algorithms

Instructor: Reza Zadeh (rezab@stanford.edu)
TA: Andreas Santucci (santucci@stanford.edu)
Problem Session, March 2017

1. Let $v_1, v_2, \ldots, v_n$ be unit vectors in $\mathbb{R}^n$. Prove that there exist $\alpha_1, \alpha_2, \ldots, \alpha_n \in \{-1, 1\}$ such that
$$||\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n||_2 \leq \sqrt{n}.$$

2. Given a complete Bipartite graph G, and given that for each $v \in V$, v has a color palette of $1 + \lceil \log_2 n \rceil$ colors, we wish to show there exists an allocation of colors to nodes such that no edge connects two nodes of the same color.[1] We have that $|L| + |R| = n$.

3. Let G be a graph with minimum degree 2. Show that G has a cycle.

4. Let G be a graph with minimum degree 3. Show G contains a cycle of even length.

5. Given a sequence p_i of stock prices on n days, we need to find the best pair of days to buy and sell. i.e. find i and j that maximizes $p_j - p_i$ subject to $j \geq i$. Give an $O(n)$ dynamic programming solution.[2]

6. Consider scheduling a sequence of jobs $J_1, J_2, \ldots, J_t$ on m machines. We wish to minimize the time it takes for the heaviest loaded machine to finish, i.e. we wish to minimize the *makespan*. Given an algorithm with competitive ratio 2 for this problem and prove the ratio.

7. Given a directed graph G and a set of terminal pairs $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, our goal is to connect as many pairs as possible using non edge intersecting paths. Provide a $\sqrt{m}$ approximation, where m is the number of edges in the graph.

8. We are given a set $I = \{1, \ldots, n\}$ of items where item $i \in I$ has size $s_i \in (0, 1]$ and a set $B = \{1, \ldots, n\}$ of bins with unit capacity. Find an assignment $a : I \to B$ such that the number of non-empty bins is minimal.

   We saw in class a 2-approximation, Any-Fit. Consider first sorting the items from largest to smallest, and show that you can obtain a 3/2 approximation to bin packing.

9. You are the sole proprietor of the company *Widgets Inc.*, a supplier of quality widgets to a large corporation in your country. You have just received a big order of w widgets from the large corporation, and are preparing a shipment from your facility, s, to their headquarters, t. Due to various idiosyncrasies in your country's postal service, the fastest way to deliver packages is to route them by hand and choose which postal hubs the package will stop in before it reaches its final destination. However, occasionally a hub's computer will crash and massively delay all shipments exiting the city. Since

[1] Notice that we don't know how many colors there are. Practical analogue: two sports teams who have a collection of shirts to choose from. How can we assign shirts to teams such that no two supporters on opposing sides wear the same shirt?

[2] We emphasize that we only have 1 stock, and one-time series. The only decision is, what pair of days should we pick to maximize profit, retrospectively.

you want to reduce the effect of such a crash, you decide that you want to minimize the number of widgets that are sent through any given hub, while still sending all w widgets. Thus, if we model our country's postal service as a graph G with a set of hubs and cities V and a set of connections between hubs E, and assume that every edge in our graph has infinite capacity (in other words, we can send as many widgets across a given connection as we like), we would like to find the smallest number k such that we can route all w widgets from s to t while ensuring that no city handles more than k widgets. Find a polynomial time algorithm to do so.

10. After your success selling widgets, you have grown your business and now sell widgets of many different sizes and shapes. Shipping these widgets requires special boxes which you have custom built by a supplier. Unfortunately, having the supplier ship you these boxes is very expensive. You notice that some of the smaller of the boxes you need fit inside some of the larger ones, and seeing an opportunity to save money, you ask your supplier to ship smaller boxes inside of larger ones. He agrees, as long as no two boxes lie side-by-side inside of a larger one– in other words, as long as no two boxes A and B lie inside box C if neither A contains B nor B contains A. (Otherwise, the two inner boxes might damage each other during shipping.) Assuming you need n different boxes and you know which of the boxes you need can fit inside others, find a polynomial time algorithm to compute the minimum number of separate shipments needed, and compute the algorithm's running time.