CME 305: Discrete Mathematics and Algorithms

Instructor: Reza Zadeh (rezab@stanford.edu)

Final Review Session – 03/20/17

1. Let $G = (V, E)$ be an unweighted, undirected graph. Let λ_1 be the maximum eigenvalue of the adjacency matrix (A) of G. Suppose Δ is the max degree of G and $\bar{d}$ is the average degree of G, then show the following:

$$\bar{d} \leq \lambda_1 \leq \Delta$$

   Note that the above also holds for weighted graphs and weighted degrees, using pretty much the same proof.

2. What are the eigenvalues of the adjacency matrix and laplacian matrix for the complete graph?

3. What are the eigenvalues of the adjacency matrix for the star graph?

   Hint: you may use the fact (without proof) that a connected graph G with maximum eigenvalue (adjacency) λ_1 is bipartite if and only if $-\lambda_1$ is also an eigenvalue.

4. Let $G = (V, E)$ be a connected, undirected graph. Let $H = (V, E')$ be a connected subgraph of G.

   (a) Show $\lambda_1(A(H)) \leq \lambda_1(A(G))$, where $\lambda_1(A(G))$ is the largest eigenvalue of the adjacency matrix associated with G.

   Hint: you may use the Perron-Frobenius thm, which says that a real square matrix with positive entries has a unique largest real eigenvalue and that the corresponding eigenvector can be chosen to have strictly positive components.

   (b) Show $\lambda_n(L(H)) \leq \lambda_n(L(G))$, where $\lambda_n(L(G))$ is the largest eigenvalue of the laplacian matrix associated with G.

5. Recall the knapsack problem: there are n items each with some value $v_1, \ldots, v_n > 0$ and weight $w_1, \ldots, w_n > 0$ and a capacity $W > 0$. Suppose $W \geq w_i$ for all i and now consider the version of knapsack where one one of each item exists. Consider the following greedy algorithm: order all items in decreasing value/weight ratio (and relabel) such that $\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \cdots \geq \frac{v_n}{w_n}$, and take the first k items that fit in the knapsack such that the next item $(k + 1)$ does not.

   (a) Show that this algorithm may be aribitrarily bad (unbounded approximation ratio).

   (b) Consider the following modified algorithm: compute the greedy solution as before and find the item of maximum value v_{i^*}. Output the maximum of the greedy algorithm and v_{i^*}. Show that this new algorithm gives a $\frac{1}{2}$-approximation.

6. The SETCOVER problem is as follows: Given a set E of elements and a collection $S_1, \ldots, S_n$ of subsets of E, what is the minimum number of these sets whose union equals E?

Let $f(e)$ be the number of sets in our collection of subsets that contain $e \in E$. Let $f = \max_{e \in E} f(e)$. Give a f-approximation algorithm to this problem.

Note that you can also come up with a $\log n$-approximation algorithm that does not depend on f. If you're interested, try doing that as well.

7. You're consulting for an e-commerce site that receives a large number of visitors each day. For each visitor i, where $i \in \{1, 2, \ldots, n\}$, the site has assigned a value v_i, representing the expected revenue that can be obtained from this customer.

   Each visitor i is shown one of m possible ads $A_1, A_2, \ldots, A_m$ as he or she enters the site. The site wants a selection of one ad for each customer so that *each* ad is seen, overall, by a set of customers of reasonably large total weight. Thus, given a selection of one ad for each customer, we will define the *spread* of this selection to be the minimum, over $j = 1, 2, \ldots, m$, of the total weight of all customers who were shown ad A_j.

   Example: Suppose there are six customers with values $3, 4, 12, 2, 4, 6$, and there are $m = 3$ ads. Then, in this instance, one could achieve a spread of 9 by showing ad A_1 to customers $1, 2, 4$, ad A_2 to customer 3, and ad A_3 to customers 5 and 6.

   The ultimate goal is to find a selection of an ad for each customer that maximizes the spread. Unfortunately, this optimization problem is NP-hard (you don't have to prove this). So instead, we will try to approximate it.

   (a) Give a polynomial-time algorithm that approximates the maximum spread to within a factor 2. That is, if the maximum spread is s, then your algorithm should produce a selection of one ad for each customer that has spread at least $s/2$. In designing your algorithm, you may assume that no single customer has a value that is significantly above the average; specifically, if $\bar{v} = \sum_{i=1}^{n} v_i$ denotes the total value of all customers, then you may assume that no single customer has a value exceeding $\bar{v}/(2m)$.

   (b) Give an example of an instance on which the algorithm you designed in part (a) does not find an optimal solution (that is, one of maximum spread). Say what the optimal solution is in your sample instance, and what your algorithm finds.

8. Consider the following problems. Show that each is NP-complete.

   (a) HITTING SET: Given a family of sets $\{S_1, S_2, \ldots, S_n\}$ and an integer b, is there a set H with b or fewer elements such that H intersects all sets in the family?

   (b) LONGEST CYCLE: Given a graph and integer k, is there a cycle with no repeated nodes of length at least k?

   (c) MAX BISECTION: Given a graph $G = (V, E)$ and integer k, does a bisection exist (i.e. $|S| = |V \setminus S| = \frac{|V|}{2}$) such that the cutsize of S is at least k?

   (d) SUBGRAPH ISOMORPHISM: Given two undirected graphs G and H, is H a subgraph of G? That is, if H has nodes $v_1, \ldots, v_n$, can you find distinct nodes $u_1, \ldots, u_n$ in G such that (u_i, u_j) is an edge in G whenever (v_i, v_j) is an edge in H?