Design and Analysis of Algorithms (XV)

Conclusion

Guoqiang Li

School of Software, Shanghai Jiao Tong University

The EXAM will be given on June. 18!

Sample of Exam: NP Problem

4. (15 points) Given a reduction from the **Clique Problem** to the following problem, which you also need to show to be a search problem.

Input a undirected graph G and a positive integer k.

Output a Clique of size k as well as an Independent Set of size k, provided both exist.

Sample of Exam: NP Problem

4. (15 points) Prove that the **Graph-Isomorphism problem** is a NP problem.

Sample of Exam: NP Problem

7. (15 points) In the **Hitting Set** problem, we are given a family of sets $\{S_1, S_2, \ldots, S_n\}$ and a budget b, and we wish to find a set H of size $\leq b$ which intersects every S_i, if such an H exists. In other words, we want $H \cap S_i \neq \emptyset$ for all i.
   Show that **Hitting Set** is NP-complete.

Sample of Exam: DP

6. (15 points) Given two strings $x = x_1x_2 \ldots x_n$ and $y = y_1y_2 \ldots y_m$, we wish to find the length of their longest common substring, that is, the largest k for which there are indices i and j with $x_ix_{i+1} \ldots x_{i+k-1} = y_jy_{j+1} \ldots y_{j+k-1}$. Show how to do this in time $O(mn)$.

Sample of Exam: DP

2. (10 points) Give an $O(n \cdot t)$ algorithm for the following task.
   Input: A list of n positive integers $a_1, a_2, \ldots a_n$, and a positive integer t.
   Question: Does some subset of the a_i's add up to t? (You can use each a_i at most once)

Sample of Exam: LP

1. (10 points) Write the dual to the following linear program.

$$\text{Max } 6x - 4z - 3$$
$$3x - y \leq 1$$
$$4y - z \leq 2$$
$$x, y, z \geq 0$$

Is the solution $(x, y, z) = (1/2, 1/2, 0)$ optimal? Write the dual program of the given linear program and find out its optimal solution.

Sample of Exam: LP

1. (10 points) Write the dual to the following linear program.

$$\text{Max } x/2 + 2y - 4$$
$$x + y \leq 3$$
$$x/2 + 3y \leq 5$$
$$x, y \geq 0$$

Find the optimal solutions to both primal and dual LPs.

Sample of Exam: LP

3. (10 points) Find the dual of the following linear program.

$$
\begin{aligned}
max\ \ & 6x_1 + 8x_2 + 5x_3 + 9x_4 + 5 \\
& 2x_1 + x_2 + x_3 + 3x_4 > 5 \\
& x_1 + 3x_2 + x_3 + 2x_4 = 3 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

Sample of Exam: LP Representation

3. (10 points) In a **facility location problem**, there is a set of facilities and a set of cities, and our job is to choose a subset of facilities to open, and to connect every city to some one of the open facilities. There is a nonnegative cost f_j for opening facility j, and a nonnegative connection cost $c_{i,j}$ for connecting city i to facility j. Given these as input, we look for a solution that minimizes the total cost. Formulate this facility location problem as an integer programming problem.

Sample of Exam: LP Representation

5. (15 points) **Set Multicover problem** is defined as an extension of **Set Cover problem**, such that each element, e, needs to be covered a specified integer number, r_e, of times. The objective again is to cover all elements up to their coverage requirements at minimum cost. We will assume that the cost of picking a set S k times is $k \cdot cost(S)$. Represent the problem as an integer linear program. Then relax to a LP and work out its dual.

Sample of Exam: LP Representation

1. (10 points) In a **Feedback Vertex Set Problem** we are given a undirected graph $G = (V, E)$ and nonnegative weights $w_v > 0$ for $v \in V$, find a set $S \subseteq V$ of minimum weight such that $G[V \setminus S]$ is a forest. Formulate this facility location problem as an integer linear programming problem, relax it to an LP, and work out its dual.

Sample of Exam: AA

5. (15 points) You are given an undirected graph. The problem is to remove a minimum number of edges such that the residual graph contains no triangle. i.e., there is no three vertices *a, b, c* such that edges *(a, b),(b, c),(c, a)* are all in the residual graph. Give a factor 3 approximation algorithm that runs in polynomial time.

Sample of Exam: AA

6. (15 points) A **Minimum Makespan Scheduling** problem is as follows:

Input processing times for n jobs, $p_1, p_2, \ldots, p_n$, and an integer m.

Output an assignment of the jobs to m identical machines so that the completion time is minimized.

Design an approximation algorithm by a greedy approach on the problem, with the approximation factor 2. And also give a tight example to show the approximation guarantee.

Sample of Exam: AA

3. (15 points) Consider the following algorithm for **Cardinality Vertex Cover**: In each connect component of the input graph execute a **depth first search (DFS)**. Output the nodes that are not leaves in the DFS tree. Show that the output is indeed vertex cover, and that it approximates the minimum vertex cover within a factor of 2.

Solution

Problem 4

Let G be the input graph, and let T denote the DFS tree constructed by our algorithm. Let $L \subseteq V(T)$ be the set of leaves and I the set of internal nodes of T.

Claim 0.1 *The set I is a valid vertex cover of G.*

Proof. Consider any edge $e \in E(G)$. If $e \in E(T)$, then clearly it is adjacent to some internal node of T and hence is covered. Now consider any edge $e = (uv) \in E(G) - E(T)$. If one of its endpoints is an internal node, then e is covered. Thus, the only remaining option is that both u and v are leaves of T. We claim that if T is a DFS tree, there are no edges joining the leaves of T.

Suppose there was an edge $e = (uv)$, and u, v were leaves of T. Consider the order in which the DFS search visited vertices of G and let u was visited before v. Since u is a leaf, we know that the DFS search did not find any unvisited neighbors of u at the time u was being visited. But that is a contradiction, because v is a neighbor of u and was visited before u. ■

Claim 0.2 *For any valid vertex cover C of G, $|I| \leq 2|C|$.*

Proof. Note that any set of vertices C that is valid vertex cover of the graph G is also a valid vertex cover of T. Hence, without loss of generality we can assume that $G = T$.

Consider any (partial) matching M in T. Note that the size of M is a lower bound on the size of any vertex cover. By showing the existence of a matching of size at least $\lceil |I|/2 \rceil$, we prove the approximation guarantee of the algorithm.

For each vertex $u \in I$, let e_u be an edge from u to an arbitrary child of u in T. Note that the edges e_u are distinct, and hence there is exactly I of them.

We split the set $M' = \{e_u \mid u \in I\}$ into two sets M_A and M_B, such that each of M_A and M_B is a matching – one of them has to be of size at least $|I|/2$.

Since T is a tree, it is a bipartite graph. Let A and B be the two sides of the bipartition. We define $M_A = \{e_u \mid u \in I \cap A\}$ and $M_B = \{e_u \mid u \in I \cap B\}$. ■

Sample of Exam: AA

5. (15 points) Consider the following two algorithms for the **knapsack problem**: (i) The greedy algorithm (pick the item with best value of $profit(i)/size(i)$); (ii) The algorithm that packs the maximum profit item.

   Prove that the algorithm that picks the better of these two solutions is a $\frac{1}{2}$-approximation for the knapsack problem.

Sample of Exam: AA

6. (15 points) A **Minimum Makespan Scheduling** problem is as follows:

Input: processing times for n jobs, $p_1, p_2, \ldots, p_n$, and an integer m.

Output: an assignment of the jobs to m identical machines so that the completion time is minimized.

We can firstly decrease order of size and send each job to the shortest queue so far.

(a) Prove that this algorithm obtains a 3/2-approximation.

(b) Give the worst gap example you can think of for this algorithm.

Sample of Exam: RA

4. (10 points) A family of hash functions $H = \{h : M \to N\}$ is said to be a **perfect hash family** if for each set $S \subseteq M$ of size $s < n$ there exists a hash function $h \in H$ that is perfect for S. The family of all possible functions from M to N is a perfect hash family.

(a) Assume $n = s$, there exist perfect hash families of size polynomial in m for $m = 2^{\Omega(s)}$.

(b) Assume $n = s$, show that any perfect hash family must have size $2^{\Omega(s)}$.

Sample of Exam: RA

8. (15 points) Consider the following probabilistic process.

Set $c = 0$
Repeat n times
    Toss a fair coin
        If heads then $c = c + 1$
        else $c = c - 1$

The sample space here is $\{H, T\}^n$, the set of all possible outcomes when tossing a coin n times, and each event has probability $1/2^n$.

Let X be the random variable defined as the value of c at the end of the process.

(1) Compute $E[X]$.

(2) For even n, show that $Pr[X = 0] \sim \sqrt{2/(\pi n)}$. What about odd n?

Hint: you will need to use Stirling's approximation, which says that

$$n! \sim \left(\frac{n}{e}\right)^n \cdot \sqrt{2\pi n}$$