Design and Analysis of Algorithms

Flow Problems and NP-Completeness

Homework IV: Turn in the solutions to problems 11 and 14 on June 8th, 2000 in class. No late or early submissions.

1. Problem 27-1, Page 625 (CLR)
2. Problem 27-2, Page 626 (CLR)
3. Problem 27-5, Page 627 (CLR)
4. **Edge-disjoint paths**

   a Given an undirected graph $G = V, E$ and two distinguished nodes, s and t, describe an algorithm that finds a maximum-sized set of edge-disjoint paths from s to t. (We say that two paths are $edge - disjoint$ if they don't share a common edge, even though they are allowed to go through the same vertex.) Give a time analysis for your algorithm in terms of $|V|$ and $|E|$.

   b Given a solution to (a), suppose we add one more edge to the graph. Give an efficient algorithm for updating the solution. Give a time analysis for your algorithm in terms of $|V|$ and $|E|$. (Your algorithm should be significantly faster than redoing the entire problem from scratch.)

5. **Row-Column sums**

   Problem You are asked to fill the entries of an $n \times n$ matrix by integers between 0 and a bound k, so that the sum of all entries in each row, and each column, comes to one of $2n$ numbers given in advance. For example, the following instance

$$\begin{array}{c} \\ 6 \\ 9 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccc} 17 & 5 & 4 \end{array} \\ \begin{pmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix} \end{array}$$

   with $k = 9$ has solution

$$\begin{array}{c} \\ 6 \\ 9 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccc} 17 & 5 & 4 \end{array} \\ \begin{pmatrix} 6 & 0 & 0 \\ 2 & 3 & 4 \\ 9 & 2 & 0 \end{pmatrix} \end{array}$$

   Formulate and solve this problem as a flow problem.

6. **Name** Job Assignment

Problem You are given a set of n jobs and a set of m machines. For each job you are given a list of machines capable of performing the job. An assignment specifies for each job one of the machines capable of performing it. The overhead for an assignment is the maximum number of jobs performed by the same machine. The job assignment problem is: given such lists and an integer $1 \leq k \leq n$, is there an assignment with overhead at most k?

7. Suppose we have a feasible flow x such that, for some nonnegative number K, there is no x-augmenting path of x-width greater than K. Prove that $f_x(s)$ is within Km of the maximum value of a feasible flow. (Hint: How would you show this for $K = 0$?) m is the number of edges in the graph.

8. Suppose that the augmenting paht algorithm always chooses an augmenting path having as few reverse edges as possible. Prove that the number of augmentations will be $O(mn)$. Give an $O(m)$ algorithm for finding such augmentation.

9. An approach to improving the augmenting path algorithm is to choose at each step an augmenting path of maximum x-width. Give a good bound on the number of augmentations. How efficiently can an augmenting path of maximum x-width be found?

10. Given a bipartite graph $G = (V, E)$ and an integer d_v for each node v, does there exist a spanning subgraph H of G such that each node has degree d_v in H. Give a good algorithm to answer this question, and also necessary and sufficient conditions for the existence of such a subgraph. A spanning subgraph of $G = (V, E)$ is a subgraph whose vertex set is V and whose edge set is a subset of E.

11. Projects $1, 2, \ldots, k$ are available to be undertaken. With each project i is associated a positive revenue r_i. Each project i requires a set S_i of resources to be available, and each resource j, $1 \leq j \leq l$, has an associated cost c_j. However if j is purchased, it is available for any projects for which it is required. Give an algorithm to choose a set of projects so that the associated revenue minus the cost of the required resources is maximized.

12. Problem 36-1, Page 961 (CLR)

13. Problem 36-2, Page 962 (CLR)

14. Problem 37-1, Page 983 (CLR)

15. Problem 37-2, Page 984 (CLR)

16. The maximum coverage problem is the following: Given a universe U of n elements, with nonnegative weights specified, a collection of subsets of U, $S_1, \ldots, S_l$, and an integer k, pick k sets so as to maximize the weight of elements covered. Show that the obvious algorithms, of greedily picking the best set in each iteration until k sets are picked, achieves an approximation factor of $(1-(1-1/k)^k) > (1 - 1/e)$.

17. Consider the following variant of metric TSP: given vertices $u, v \in V$, find a minimum cost simple path from u to v that visits all vertices. First give a factor 2 approximation algorithm for this problem, and then improve it to factor 3/2.

18. Give a factor 2 approximation algorithm for the following problem: Given a directed graph $G = (V, E)$ with nonnegative edge costs, and a partitioning of V into two sets Senders and Receivers, find a minimum cost subgraph such that every Receiver vertex has a path to a Sender vertex.

19. The low degree spanning tree problem is as follows: Given a graph G and an integer k, does G contain a spanning tree such that all vertices in the tree have degree at most k (obviously, only tree edges count towards the degree)?

    Prove that the low degree spanning tree problem is NP-hard with a reduction Hamiltonian path.

    Now consider the high degree spanning tree problem: Given a graph G and an integer k, does G contain a spanning tree whose highest degree vertex is has degree least k? Give an efficient algorithm to solve the high degree spanning tree problems and an analysis of its time complexity.