CS 202 Homework 3
Dynamic Programming and Network Flow
Due June 5, 2008

General directions: Algorithms may be described at "high level" without actual code. You may use any lower bound, algorithm or data structure from the text or in class, and their correctness and analysis, but be careful. For example, if you construct a graph with n^2 nodes and n^3 edges and then run Dijkstra's algorithm on the resulting graph, the total run-time is $O(n^3 logn)$,because Dijkstra's algorithm is $O(ElogV)$. If you want to use the correctness of Dijkstra's algorithm on the graph, you must check that edge weights are positive, since otherwise the proof that Dijkstra's algorithm gives shortest paths does not go through. If a problem has multiple size parameters, you should express the run-time as a function of all relevant parameters; e.g., saying maximum bi-partite matching takes time $O(|V||E|)$ is more accurate than to say it takes time $O(V^3)$, although both are correct.

For the first four problems, you must prove correctness and give a time analysis. For some problems, correctness may be trivial; for others the analysis may be trivial. If it is trivial, you do not need to go into detail, but you should at least give a one or two line explanation. (Conversely, if you only give a one or two line explanation, I will view this as implicitly claiming that it is trivial.)

Each problem is worth 10 points. Grading may be based on any of the following that are relevant for the problem: the efficiency of your algorithm; the correctness and proof thereof; and the time analysis. The number of points depending on each part is given after the problem, as well as a ballpark estimate of the time analysis for my solutions of the problem.

Grade maximization Problem 20 on pp. 329-330.

Number of rectangular prisms of given diagonal . Consider a *rectangular prism* of dimension d, with sides $s_1, ..s_d$. Then the diagonal will have length D so that $D^2 = \sum_i s_i^2$. Give an algorithm which, given integers d and m, returns the number of prisms with *integer-valued* sides $s_1, ..s_d$ and with diagonal D so that $D^2 = m$. Your algorithm should be poly-time in the values m and d. (7 points correct poly-time algorithm, 3 pts efficiency. My best time is $O(md)$.)

Job allocation 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.

The best method for solving this problem will depend on the relationship between n and m. Your algorithm should be correct and polynomial time for any such values, but give an efficient version of your algorithm and time analysis in the case when $m = n/\log n$. (6 pts correct poly-time algorithm, 4 pts. efficiency.)

Finding Bottleneck edges Consider a directed graph G with two specified nodes s and t. Let the *path number* be the maximum size of a set of paths from s to t that are pairwise edge-disjoint, i.e. no two paths in the set share an edge.

a. Give an efficient algorithm to determine the path number in a graph.

b. An edge is a *bottleneck* if removing the edge decreases the path number. Give an efficient algorithm to find all the bottlenecks.

Implementation of Independent Set Consider a greedy heuristic for independent set that selects the lowest degree node, puts it in the set, and deletes it and its neighbors and repeats. Implement this greedy heuristic, and a correct backtracking algorithm for Maximum Independent Set. For as many powers of 2 as you can, starting with $n = 64$, and for $p = 1/2, 1/4$ and $1/8$, run both algorithms on random graphs where edges are independently placed between nodes with probability p. What are the sizes of the greedy heuristic sets compared to the optimal independent sets found by the backtracking algorithm? Give a table of average values, and the maximum ratio.