

## Homework Assignment 5

November 11, 2003

Due Date: November 25

## Core Problems

1. (10 pts) In this problem you will analyze an approximation algorithm for the knapsack problem. Let the input be  $v_1, \dots, v_n$  (the item values),  $w_1, \dots, w_n$  (the item weights), and  $W$  (the knapsack capacity). The goal is to find a set of items  $S$  such that  $\sum_{i \in S} w_i \leq W$  and  $\sum_{i \in S} v_i$  is maximized. Here is an approximation algorithm. First sort the items so that

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \dots \geq \frac{v_n}{w_n}$$

We now use the following variation of the greedy algorithm. Let  $S_{greedy}$  be the answer computed by the greedy algorithm. Namely,  $S_{greedy}$  chooses the first  $k$  objects that fit. That is,  $k$  satisfies

$$\sum_{i=1}^k w_i \leq W \quad \text{and} \quad \sum_{i=1}^{k+1} w_i > W.$$

The approximation algorithm returns the better of  $S_{greedy}$  or the single item  $k+1$ . Prove the best approximation ratio you can for this algorithm. Some useful expressions to consider are  $v_1 + v_2 + \dots + v_k + v_{k+1}$  and the value of the approximate solution:  $\max\{v_1 + \dots + v_k, v_{k+1}\}$ .

2. (20 pts) In this problem we consider the relationship between the traveling salesman problem with an unrestricted cost function (TSP) and the traveling salesman problem when the cost function must satisfy the triangle inequality (TSPWTI).

Here is a polynomial time reduction from the optimization version of TSP to the optimization version of TSPWTI. Let  $G$  be the TSP input in which there are vertices  $V$  and for each pair  $u, v \in V$ , there is an edge with weight  $w(u, v)$ . Let  $W$  be the largest edge weight. We construct the TSPWTI input  $G'$  as follows.  $G'$  has the same vertices as in  $G$ . In  $G'$  for each pair  $u, v \in V$ , there is an edge with weight  $w'(u, v) = W + w(u, v)$ .

- (3 pts) Argue that the edge weights in  $G'$  satisfy the triangle inequality. That is for an arbitrary three vertices  $x, y, z$  prove that  $w'(x, y) + w'(y, z) \geq w'(x, z)$ .
- (3 pts) Since  $G$  and  $G'$  have the same set of vertices a tour  $T$  in  $G$  is also a tour in  $G'$ . Prove that  $T$  is an optimal tour in  $G$  if and only if it is an optimal tour in  $G'$ .
- (9 pts) Suppose one uses this reduction to create the following approximation algorithm  $A$  for TSP. Given TSP input  $B$ , apply the given transformation to create  $G'$ . Now apply the 2-approximation given in class (and in the text) to the the TSPWTI instance  $G'$ . Output the tour computed by this approximation algorithm (for  $G'$ ) as the tour for  $G$ . Prove the best ratio bound you can for this approximation algorithm for the general TSP problem.
- (5 pts) Assuming  $P \neq NP$ , we proved that for any constant  $c \geq 1$  there is no polynomial time approximation algorithm with ratio  $c$  for the general TSP problem. Explain why your answer for part (c) does not contradict this result.

ALTERNATE PROBLEM 2: Implement (in any language) the fully polynomial-time approximation scheme (PTAS) given in the text for the optimization version of the subset sum problem. You are to output the actual solution (i.e. the list of the elements selected) along with their sum.

On the web page is a sample data file. The first line contains  $n$ , the cardinality of the set  $S$  and the integer  $t$ . Then second line contains  $n$  integers (the elements of  $S$ ).

Provide your code as well as your output for the data set on the course web page for  $\epsilon = .2$ ,  $\epsilon = .1$  and  $\epsilon = .01$ .

3. (10 pts) Give the best lower bound you can for the problem of finding the median among the  $2n + 1$  elements in the two sorted arrays  $A[1] < A[2] < \dots < A[n] < A[n + 1]$  and  $B[1] < B[2] < \dots < B[n]$  using a comparison-based algorithm that can only learn about the array elements by posing questions of the form is  $A[i] < B[j]$  for  $1 \leq i \leq n + 1$  and  $1 \leq j \leq n$ .
4. (15 pts) Consider the problem of determining whether an undirected graph with  $n$  vertices has a vertex of degree 2 or more. Assume that the algorithm is restricted to ask questions of the form “Is there an edge between  $i$  and  $j$ ?”. Give the best adversary argument you can on the number of such questions required to determine if there is a vertex in the graph with degree 2 or more.

### Advanced Problems (Required only for CS 539T students)

5. (15 pts) Consider the problem of determining whether an undirected graph  $G$  with  $2^k$  vertices is connected (i.e. whether there is an undirected path between each pair of vertices). Assume that the algorithm is restricted to ask questions of the form “Is there an edge between  $i$  and  $j$ ?”. Give the best adversary argument you can on the number of such questions required to determine if  $G$  is connected. You should go well beyond the solution for Practice Problem 2.

*Hint: Extend the adversary argument given in the practice problem solutions for Problem 2. Think recursively.*

### Challenge Problem (Optional for all students)

6. Using an adversary argument to show that *any* comparison-based algorithm to find the second smallest of  $n$  elements must make at least  $n + \lceil \lg n \rceil - 2$  comparisons. (Although I’m not asking you to find it, there is a comparison-based algorithm that makes at most  $n + \lceil \lg n \rceil - 2$  comparisons.)

We’ll just record a \* (done correctly) or  $\checkmark$  (very close) as well as giving you comments on what you submit.