CS570

Analysis of Algorithms

Fall 2006

Final Exam

Name: ________________________

Student ID: ____________________

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	15	
Problem 4	15	
Problem 5	15	
Problem 6	20	

Note: The exam is closed book closed notes.

1) 20 pts
Mark the following statements as **TRUE**, **FALSE,** or **UNKNOWN**. No need to provide any justification.

[TRUE/FALSE]
P is the class of problems that are solvable in polynomial time.

[TRUE/FALSE]
NP is the class of problems that are verifiable in polynomial time

[TRUE/FALSE]
It is not known whether P ≠ NP

[TRUE/FALSE]
If an NP-complete problem can be solved in linear time, then all the NP complete problems can be solved in linear time

[TRUE/FALSE]
Maximum matching problem in a bipartite graph can be efficiently solved thru the solution of an equivalent max flow problem

[TRUE/FALSE]
The following recurrence equation has the solution $T(n) = \Theta(n \cdot log(n^2))$.

$$T(n) = 2T(\frac{n}{2}) + 3 \cdot n$$

[TRUE/FALSE/UNKNOWN]
If a problem is not in P, it should be in NP-complete

[TRUE/FALSE/UNKNOWN]
If a problem is in NP, it must also be in P.

[TRUE/FALSE/UNKNOWN]
If a problem is NP-complete, it must not be in P.

[TRUE/FALSE]
Integer programming is NP-Complete

2) 15 pts

A cargo plane can carry a maximum of 100 tons and a maximum of 60 cubic meters of cargo. There are three materials that need to be carried, and the cargo company may choose to carry any amount of each, up to the maximum amount available of each.

- Material 1 has density 2 tons/cubic meter, maximum available amount 40 cubic meters, and revenue $1000 per cubic meter.
- Material 2 has density 1 tons/cubic meter, maximum available amount 30 cubic meters, and revenue $1200 per cubic meter.
- Material 3 has density 3 tons/cubic meter, maximum available amount 20 cubic meters, and revenue $12000 per cubic meter.

Write a linear program which optimizes revenue within the given constraints.

3) 15 pts

   Given two sorted arrays a[1,…, n] and b[1,…,n], present an O(log n) algorithm to search the median of their combined 2n elements.

4) 15 pts

Present an efficient algorithm for the following problem.

Input: n positive integers $a_1, a_2, a_3, \ldots a_{n-1}, a_n$ and number **t**

Task: Determine if there exists a subset of the $a_i's$ whose sum equals **t**?

5) 15 pts

There are M faculties and N courses. Every faculty chooses 2 courses based on his/her preference. A faculty can teach zero, one course or two courses and a course can be taught by one faculty only.

Find a feasible course allocation if one exists (A feasible course allocation allows a faculty to teach only the courses he/she prefers).

2) 20 pts

In a weighted graph, the length of path p=<$v_0, v_1, \ldots, v_k$> is the sum of the weights of its constituent edges. A path is simple if all vertices in the path are distinct. Let D denote an algorithm for the decision version of the longest path problem. The input consists of a weighted, connected undirected graph G=(V,E) along with an integer k. The output is "yes" if and only if G has a simple path of length k or more.

a) Give an algorithm that uses D to compute the length of the longest path. (You don't have to output the actual path.)

b) (True / False) The longest path problem is NP-complete. If True prove it. If False disprove it.

CS570 FALL 2006 FINAL SOLUTION

Problem 1: 1.True 2.True 3.False 4.False 5.True 6.True 7.False 8.Unknown 9.Unknown 10.True

Problem 2:

Solution 1): Assume that the weight of material 1, material 2 and material 3 carried on the cargo plane are x_1,x_2 and x_3 respectively. Clearly $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$. Then the total revenue is $1000 * x_1/2 + 1200 * x_2/1 + 12000 * x_3/3$. As the cargo plane can carry a maximum of 100 tons and a maximum of 60 cubic meters of cargo, we have $x_1 + x_2 + x_3 \leq 100$, $x_1/2 + x_2/1 + x_3/3 \leq 60$; As Material 1 has maximum available amount 40 cubic meters, we have $x_1/2 \leq 40$;As Material 2 has maximum available amount 30 cubic meters, we have $x_2/1 \leq 30$; As Material 3 has maximum available amount 20 cubic meters, we have $x_3/3 \leq 20$; Hence the linear program which optimize revenue is:

$$\max(1000 * x_1/2 + 1200 * x_2/1 + 12000 * x_3/3)$$

$$\text{subject to: } \begin{cases} x_1 \geq 0, \\ x_2 \geq 0, \\ x_3 \geq 0, \\ x_1 + x_2 + x_3 \leq 100, \\ x_1/2 + x_2/1 + x_3/3 \leq 60, \\ x_1/2 \leq 40, \\ x_2/1 \leq 30, \\ x_3/3 \leq 20. \end{cases}$$

Solution 2): Assume that the amount of material 1, material 2 and material 3 carried on the cargo plane are x_1,x_2 and x_3 respectively. Clearly $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$. Then the total revenue is $1000 * x_1 + 1200 * x_2 + 12000 * x_3$. As the cargo plane can carry a maximum of 100 tons and a maximum of 60 cubic meters of cargo, we have $x_1 + x_2 + x_3 \leq 60$, $x_1 * 2 + x_2 * 1 + x_3 * 3 \leq 100$; As Material 1 has maximum available amount 40 cubic meters, we have $x_1 \leq 40$;As Material 2 has maximum available amount 30 cubic meters, we have $x_2 \leq 30$; As Material 3 has maximum available amount 20 cubic meters, we have $x_3 \leq 20$; Hence the linear program which optimize revenue is:

$$\max(1000 * x_1 + 1200 * x_2 + 12000 * x_3)$$

Date: Dec 8, 2006.

$$\text{subject to: } \begin{cases} x_1 \geq 0, \\ x_2 \geq 0, \\ x_3 \geq 0, \\ x_1 + x_2 + x_3 \leq 60, \\ x_1 * 2 + x_2 * 1 + x_3 * 3 \leq 60, \\ x_1 \leq 40, \\ x_2 \leq 30, \\ x_3 \leq 20. \end{cases}$$

Problem 3: The problem is essentially the same as Problem 1 in Chapter 5, which is assigned in HW#5.

We denote $A[1, n], B[1, n]$ as the sorted array in increasing order and $A[k]$ as the $k-th$ element in the array. For brevity of our computation we assume that $n = 2^s$. First we compare $A[n/2]$ to $B[n/2]$. Without loss of generality, assume that $A[n/2] < B[n/2]$, then the elements $A[1], ..., A[n/2]$ are smaller than n elements, that is,$A[n/2], ..., A[n]$ and $B[n/2], ..., B[n]$. Thus $A[1], ..., A[n/2]$ can't be the median of the two databases. Similarly, $B[n/2], ..., B[n]$ can't be the median of the two databases either. Note that m is the median value of A and B if and only if m is the median of $A[n/2], ..., A[n]$ and $B[1], ..., B[n/2]$(We delete the same number of numbers that are bigger than m and smaller than m), that is, the $n/2$ smallest number of $A[n/2], ..., A[n]$ and $B[1], ..., B[n/2]$. Hence the resulting algorithm is:

Algorithm 1 Median-value($A[1, n], B[1, n], n$)
if $n = 1$ then
    return min(A[n],B[n]);
else if $A[n/2] > B[n/2]$
    Median-value($A[1, n/2], B[n/2, n], n/2$)
else if $A[n/2] < B[n/2]$
    Median-value($A[n/2, n], B[1, n/2], n/2$)
end if

For running time, assume that the time for the comparison of two numbers is constant, namely, c, then $T(n) = T(n/2) + c$ and thus $T(n) = c \log n$. Hence our algorithm is $O(\log n)$.

Problem 4: This problem is actually equivalent to solving the Subset Sum problem: Given n items $\{1, , n\}$ and each has a given nonnegative weight a_i, we want to maximize $\sum a_i$ given the condition that $\sum a_i = t$. If the maximum value returned is t, then the output is yes, otherwise false. The recursive relation is:

$$\begin{cases} OPT(i, t) = OPT(i-1, t), \text{if } t < a_i; \\ OPT(i, t) = \max(OPT(i-1, t), a_i + OPT(i-1, t-a_i)), \text{otherwise;} \end{cases}$$

Time complexity: Note the we are building a table of solutions M, and we compute each of the values $M[i, t]$ in $O(1)$ time using the previous values, thus the running time is $O(nt)$.

Or The Subset-Sum algorithm given in the textbook is $O(nt)$, hence the running time of our algorithm is $O(nt)$.

Problem 5: This problem can easily be reduced into max flow problem. We denote the M faculty as $\{x_1, ..., x_M\}$ and N courses as $\{y_1, ..., y_N\}$. The reduction is as follows: add source s, sink t in the graph. For each x_i, add edge $s \rightarrow x_i$. The capacity lower bound of these edges is 1 and the capacity upper bound is 2.For each

y_i, add edge $y_i \to t$. The capacity of each edge is 1. For each faculty x_i, if x_i prefer course y_{i_1} and y_{i_2}, add edges $x_i \to y_{i_1}$ and $x_i \to y_{i_2}$ in the graph. The capacity of each edge is 1. To find a feasible allocation, we just need to solve the max flow problem in the constructed graph to find a max flow with value N.

Problem 6:

a)To compute the length of the longest path, we just need to find a integer k such that $D(G,k) = Yes$ and $D(G,k+1) = No$. Hence the algorithm is:

$k = 0$;
while $D(G,k) = Yes$
    $k++$;
return k;

b)True. The longest path problem is NP-complete. The proof is as follows: Clearly given a path we can easily check if the length is k in polynomial time, hence the problem is NP. To prove that it is NP-complete, we prove that even the simplest case of longest path problem is NP-complete. The simplest case in our longest path problem is that the weight of each edge is 1. We use a simple reduction from HAMILTON PATH problem: Given an undirected graph, does it have a Hamilton path, i.e, a path visiting each vertex exactly once? HAMILTON PATH is NP-complete. Given an instance $G' =< V', E' >$ for HAMILTON PATH, count the number $|V'|$ of nodes in G' and output the instance $G = G', K = |V'|$ for LONGEST PATH. Obviously, G' has a simple path of length $|V'|$ if and only if G' has a Hamilton path. Therefore, if we can solve LONGEST PATH problem, we can easily solve HAMILTON PATH problem. Hence LONGEST PATH Problem is NP-complete.