COMP 465/6651 Design and Analysis of Algorithms
Fall 2004
Final exam
Solutions

1. [20 points] Consider a bipartite graph G with n left vertices and n right vertices; each edge e of G has been assigned a positive weight $w(e)$ in such a way that

> for each of the $2n$ vertices of G,
> the total weight of edges incident with this vertex equals 1.

(Edge e is said to be incident with vertex v if v is one of the two endpoints of e.) Use the König-Hall theorem to prove that G has a matching of size n.

(Hint: What is the total weight of all the edges of G?)

Solution: Since each edge of G is incident with precisely one of the n left vertices and since the total weight of edges incident with each of these n vertices equals 1, the total weight of all the edges of G equals n. If C is a vertex-cover in G then each edge of G is incident with at least one of the vertices in C; since the total weight of edges incident with each of these $|C|$ vertices equals 1, the total weight of all the edges of G is at most $|C|$. It follows that the smallest vertex-cover in G has size at least n (in fact, precisely n, but this is beside the point); this fact and the König-Hall theorem together guarantee that the largest matching in G has size at least n (in fact, precisely n, but this is beside the point).

2. [20 points] Present, in our usual pseudocode style, a dynamic programming algorithm that, given positive integers

$$a_1, a_2, \ldots, a_n, b, c_1, c_2, \ldots, c_n,$$

finds the largest of all the values of $\sum_{i=1}^{n} c_i x_i$ where

$$\sum_{i=1}^{n} a_i x_i \leq b \text{ and } x_i \in \{0, 1\} \text{ for all } i.$$

Express the running time of your algorithm in the asymptotic Θ notation.

One solution:

for $t = 0$ to b **do** $\textsc{Best}(0, t) = 0$ **end**
for $k = 1$ to n
do **for** $t = 0$ to b
 do **if** $a_k \le t$ and $\textsc{Best}(k-1, t-a_k) + c_k > \textsc{Best}(k-1, t)$
 then $\textsc{Best}(k, t) = \textsc{Best}(k-1, t-a_k) + c_k$
 else $\textsc{Best}(k, t) = \textsc{Best}(k-1, t)$
 end
 end
end
return $\textsc{Best}(n, b)$;

The running time of this algorithm is in $\Theta(nb)$.

3. [15 points] Reduce HAMILTONIAN PATH to HAMILTONIAN CYCLE.

One solution: Given a graph G, add a new vertex and make it adjacent to all the old vertices of G. Now G has a Hamiltonian path if and only if the new graph has a Hamiltonian cycle.

4. [15 points] Prove that the following problem is NP-complete:

HALF-CLIQUE
Input: An undirected graph with $2k$ vertices.
Question: Does the input graph contain a clique on k vertices?

One solution: Clearly, HALF-CLIQUE belongs to NP. To prove that HALF-CLIQUE is NP-hard, we shall reduce CLIQUE to HALF-CLIQUE: given an undirected graph G and a positive integer t, we shall construct an undirected graph H with $2k$ vertices such that

G contains a clique on t vertices if and only if
H contains a clique on k vertices.

For this purpose, let n denote the number of vertices of G. If $n = 2t$, then we set $H = G$. If $n < 2t$, then we let H consist of G plus $2t - n$ new vertices that have no neighbours.If $n > 2t$, then we let H consist of G plus $n - 2t$ new vertices that are adjacent to each other as well as to the n old vertices.

5. [15 points] Prove that the following problem is NP-complete:

FOLDING RULER
Input: A sequence of positive integers $a_1, a_2, \ldots, a_n$
and a positive integer b.
Question: Can a ruler whose sequence of rod lengths is $a_1, a_2, \ldots, a_n$
fit into a pocket of depth b?

Hint: Can a ruler whose sequence of rod lengths is

$$1000, 500, 14, 9, 11, 16, 10, 500, 1000$$

fit into a pocket of depth 1000?

One solution: Clearly, FOLDING RULER belongs to NP. To prove that FOLDING RULER is NP-hard, we shall reduce FIFTY-FIFTY to FOLDING RULER: given positive integers $c_1, c_2, \ldots, c_m$, we shall construct a sequence of positive integers $a_1, a_2, \ldots, a_n$ and a positive integer b such that

> there is a subset I of $\{1, 2, \ldots, m\}$ such that $\sum_{i \in I} = \sum_{i \notin I}$
> if and only if a ruler whose sequence of rod lengths is $a_1, a_2, \ldots, a_n$
> can fit into a pocket of depth b.

For this purpose, we choose a big even integer b ($b > \sum_{i=1}^{m} a_i$ is big enough) and we let the sequence $a_1, a_2, \ldots, a_n$ be

$$b, b/2, c_1, c_2, \ldots, c_m, b/2, b.$$

6. [15 points] Where is the fallacy in the following argument?

> "In Problem 1 of Homework assignment 4, we have designed a dynamic programming algorithm that solves FOLDING RULER in time $\Theta(nb)$, and so we have proved that FOLDING RULER belongs to P. Now (in the preceding problem in this exam) we are asked to prove that FOLDING RULER is NP-complete; if we succeed, then we will have proved that P=NP."

The solution: Proving that FOLDING RULER can be solved in time $\Theta(nb)$ does not prove that FOLDING RULER belongs to NP: b is not polynomially bounded by the size of the input.