Department of Computer Science Fall 2011 New Mexico State University

Ph.D. Qualifying Exam: Analysis of Algorithms

This is a closed book exam. The total score is 100 points. Please answer all questions.

(30 points) 1. Give a linear time algorithm to determine if an undirected graph $G = (V, E)$ is *bipartite*. A graph is bipartite if the set of vertices V can be divided into two nonempty subsets V_1 and V_2 such that there is no edge between any two vertices in the same subset.

It may be helpful to represent graph G by an adjacency list.

Solution:

This problem can be solved by performing a labeled depth-first search on G. During the DFS, we will attempting to label each vertex a label of -1 or 1, so that the two vertices of each edge may have opposite labels, indicating two different partitions.

1. For the root vertex of each DFS tree, pick a label arbitrarily.
2. Every time a vertex v is discovered by u, let label(v) = -label(u).
3. When a back edge is detected, if label(v) $\neq$ -label(u), then G is not bipartite.
4. If no such back edge exists in the graph, then G is bipartite.

2. Let a steel sheet have a rectangular shape of size $X \times Y$. We assume both X and Y are positive integers. We can use the sheet to produce some of a list of n items of smaller rectangular pieces of $x_i \times y_i$ (both positive integers) with a price c_i $(i = 1, \ldots, n)$.

The sheet can only be cut horizontally or vertically into two pieces each time. You can make more than one number of the same size items.

(30 points) (a) Give an efficient algorithm to determine the maximum profit that can be made by cutting the steel sheet into pieces.

Solution:

Let $p(x, y)$ be the total profit can be made on a steel sheet of size $x \times y$. We use dynamic programming based on the following recurrence equation:

$$p(x,y) = \begin{cases} 0 & x \cdot y = 0 \\ \max \begin{cases} \max\limits_{1 \le u < x} & p(u,y) + p(x-u,y) \\ \max\limits_{1 \le v < y} & p(x,v) + p(x,y-v) \\ \max\limits_{i, x_i = x, y_i = y} & c_i \end{cases} & x \cdot y \neq 0 \end{cases}$$

(10 points) (b) Determine the running time of your algorithm.

Solution: There are $X \cdot Y$ many subproblems, each taking time $\Theta(X+Y)+\Theta(n)$ to compute. Therefore, the total time is

$$\sum_{x=1}^{X}\sum_{y=1}^{Y}\left(\sum_{u=1}^{x}\text{constant}+\sum_{v=1}^{y}\text{constant}+n\right)=\Theta(XY(X+Y+n))$$

(30 points) 3. We use a variable number of bits to represent each number from 1 to n in binary, i.e.,

$$1 \equiv 1_2,\ 2 \equiv 10_2,\ 3 \equiv 11_2,\ 4 \equiv 100_2,\ 5 \equiv 101_2,\ \ldots$$

What is the tight order of the number of bits for the factorial $n!$ in binary, using the Θ asymptotic notation? Please include both upper and lower bound analysis.

Solution: For number i, we use exactly $\lfloor 1+\lg i \rfloor$ bits.

Lower bound: For the factorial $n!$, we will thus use at least

$$\sum_{i=1}^{n}\lfloor 1+\lg i \rfloor \text{ bits} = \Omega(n \lg n)$$

Upper bound: As number i will not need more than $1+\lg n$ bit, we will have no more than

$$n(1+\lg n) \text{ bits} = O(n \lg n)$$

for $n!$.

Therefore, the number of bits for $n!$ in binary is $\Theta(n \lg n)$.