

Theory Qualifier Exam, 2011

ID Number: _____

1. You are given n books, B_1, B_2, \dots, B_n that need to be stored on shelves. B_i is t_i inches thick and h_i inches high. The books must be placed *in order* on successive shelves so as to minimize the overall height of the shelves (each shelf is as tall as its highest book; the overall height is the sum of the heights of the individual shelves). It would clearly be best if all the books could be stored on one shelf—then the height would be $\max_{1 \leq i \leq n} h_i$ inches. However, no shelf can be longer than L inches, so you need to find break points $i_0 = 1, i_1, i_2, \dots, i_m = n + 1$ so that the books fit on the shelves,

$$\sum_{i_j \leq i < i_{j+1}} t_i \leq L,$$

while minimizing the sum of the heights of the highest book on each shelf,

$$\sum_{j=0}^{m-1} \max_{i_j \leq i < i_{j+1}} h_i.$$

That is, the books are put onto m shelves: B_1, \dots, B_{i_1-1} on shelf one, $\dots, B_{i_{m-1}}, \dots, B_n$ on shelf m , so that in general, $B_{i_{j-1}}, \dots, B_{i_j-1}$ are stored on shelf j , $j = 1, 2, \dots, m$.

- Using the *Principle of Optimality* of dynamic programming, write a recursive expression for the overall minimum height of the book shelves; be sure to give the necessary initial values.
- Analyze the cost (as a function of n) of evaluating your recursive expression *without memoization* to determine the minimum height of the book shelves.
- Describe precisely how memoization could be done to avoid duplicate solution of subproblems.
- Assuming memoized code, analyze the time required to determine the minimum height of the book shelves.
- What additional memoization is needed to determine the break points between shelves?

2. The Fibonacci Numbers F_n , for $n \geq 0$ is defined as follows. $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, and

$$F_n = F_{n-1} + F_{n-2}, \quad \text{for } n \geq 3$$

Assume that, given any two integers x and y , we can compute their summation $x + y$ and their product $x \cdot y$ in a constant time. Design a method to compute the value of F_n in time $\Theta(\log n)$. You need to give a pseudocode of your method and also justify that your algorithm indeed runs in time $\Theta(\log n)$.

Hint: Notice the following formula is true for Fibonacci numbers, when $n \geq 2$:

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix}$$

3. Suppose we have a graph that has $O(n + k)$ edges where k is a constant. Design an $O(n)$ algorithm to find a minimum spanning tree in this graph. Hint: There are $k + 1$ extra edges in the graph.

4. Suppose we have m teachers in the department, and n courses to be offered. Each course can be taught by a specified subset of teachers. You are required to identify the minimum number of teachers required so that each and every course can be taught by some teacher in the the set you identified. Can you solve this problem efficiently? You can use the fact that the decision version of the following problems are NP-Complete: *Travelling Salesman Problem*, *Vertex Cover*, *Independant Set*, *Set Cover*, *Dominating Set*.