Homework #12
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2014

Due on: Tuesday, April 29th, 5pm
Format: Please start each problem on a new page.
Where to submit: On blackboard, under student assessment
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

April 22, 2014

Note: This homework is entirely optional, but please submit it before 5pm on Tuesday, April 29th if you wish to receive feedback.

Problem 1 (Optional)

Consider a team of gamblers $P = \{p_1, \ldots, p_n\}$ in a casino. After a day of gambling, each player p_i had won d_i dollars. The casino has an additional rule: if there is a subset of players who win $A = n^2$ dollars in total, then the entire group gets an additional 10000 dollars. Design a dynamic programming algorithm that computes the total score of the team. Your algorithm should have a polynomial running time (in terms of n). What will happen to the performance of your algorithm if $A = 2^n$?

Problem 2 (Optional)

Let π be a permutation of $[n] = \{1, 2, \ldots, n\}$. Let A be an array of size m with elements from $[n]$. A is "symmetric" with respect to π if $A[1] = \pi(A[m])$, $A[2] = \pi(A[m-1])$, ..., $A[i] = \pi(A[m-i+1])$. Given π and an array B of size N with elements from $\{1, \ldots, n\}$, our goal is to find a longest subarray $B[i, \ldots, j]$ that is symmetric with respect to π. Design a dynamic programming algorithm that solves this problem. Full credit will be given if your algorithm works in polynomial time, solves the problem and your analysis is correct and rigorous.

Problem 3 (Optional)

Suppose you have n boxes $b_1, b_2, \ldots, b_n$. Each box has a height, width and depth. For box b_i, let us denote the height, width and depth of box b_i as h_i, w_i and d_i, respectively. We would like to build the tallest possible tower of boxes. A tower of boxes must obey the property that box b_i can be stacked on top of b_j only if the dimensions of the base of b_i are strictly larger than the dimensions of the base of b_j. Of course, you are permitted to rotate a box however you like, so, for example, you can rotate box b_i so that its base has dimension $h_i \times d_i$.

Give a dynamic programming algorithm to build the tallest possible tower of boxes.

Bonus challenge problem: give an algorithm that works for hyper-boxes– boxes in d-dimensional space. A hyperbox in d dimensions has d side lengths, $\ell_1, \ell_2, \ldots, \ell_d$, so your input would be n hyper-boxes, each of which is described by d numbers.

Optional exercises

Solve the following problems and exercises from CLRS: 15-7, 15-10.