CS344: MidTerm (sample)
Thursday Oct 26, 11:45–12:45.

October 30, 2017

1. Dynamic Programming. Here is 2D knapsack problem. As before, there are n items, each with a integer weight w_i and an integer value v_i , and a weight limit W. In addition, each item has a size s_i and we have a size limit S (imagine there is both a weight limit on what you can carry and a size limit of what you can fit in the knapsack). We want the subset of items of maximum value such that their weight is at most W and their size is at most S.

2. Given a tree with nodes numbered 1 to n, and children of node i having weight $w_1, w_2, \ldots, w_{d_i}$ where d_i is degree of i and w_j's are weights in the range $[1, m]$. Store them in one table of size $O(n)$ so that you can answer query (j, k) in $O(1)$ time, returning the pointer to child of node j of weight k.

3. We are given a text string t of length $2n$. We are also given a pattern string p of length n, but the pattern has some positions marked "wildcards" where we dont care what letter matches. For example $p = a\phi b\phi$ means p can match $aXbY$ or $axbb$ or .... Say p has at most k such wildcards ϕ. Design a Karp-Rabin fingerprint method to find all the places where p matches t. How much running time does the algorithm take?

4. Solve the recurrence $T(n) = T(n^{1/4}) + T(n^{1/2}) + \log n$.

5. Analyze the median finding algorithm from the class by dividing the array into groups of $2\ell + 1$ items, and assuming the median of each group can be found in $O(\ell^2)$ time.