Lindsey Bieda and Joe Frambach
Dynamic Programming Problems
10.05.2011

20. The input to this problem is two sequences $T = t_1, \ldots, t_n$ and $P = p_1, \ldots, p_k$ such that $k = n$, and a positive integer cost c_i associated with each t_i. The problem is to find a subsequence of T that matches P with maximum aggregate cost. That is, find the sequence $i_1 < \ldots < i_k$ such that for all j, $1 = j = k$, we have $t_{i_j} = p_j$ and $\sum_{j=1}^{k} c_{i_j}$ is maximized.
So for example, if $n = 5$, $T = XY\ XXY$, $k = 2$, $P = XY$, $c_1 = c_2 = 2$, $c_3 = 7$, $c_4 = 1$ and $c_5 = 1$, then the optimal solution is to pick the second X in T and the second Y in T for a cost of $7 + 1 = 8$.

(a) Give a recursive algorithm to solve this problem. Then explain how to turn this recursive algorithm into a dynamic program.

wss(i,j):
    if i = 0 or j = 0:
        return 0
    if $T_i = P_j$:
        return max(v_i + wss(i-1,j-1), wss(i-1,j))
    else:
        return wss(i-1,j)

(b) Give a dynamic programming algorithm based on enumerating subsequences of T and using the pruning method.

(c) Give a dynamic programming algorithm based on enumerating subsequences of P and using the pruning method.