Lindsey Bieda and Joe Frambach
Dynamic Programming Problems
10.05.2011

21. The input to this problem is a sequence of n points $p_1, \ldots, p_n$ in the Euclidean plane. You are to find the shortest routes for two taxis to service these requests in order. Let us be more specific. The two taxis start at the origin. If a taxi visits a point p_i before p_j then it must be the case that $i < j$. (Stop and think about what this last sentence means.) Each point must be visited by at least one of the two taxis. The cost of a routing is just the total distance traveled by the first taxi plus the total distance traveled by the second taxi. Design an efficient algorithm to find the minimum cost routing.

Tree description:

The tree nodes are indexed by [*taxi* 1*'s position*, *taxi* 2*'s position*]. This scheme brings the total space requirement to $O(n^2)$.

Pruning Rule:
1) For all values the same level with equal indices, prune all but the shortest. There will be $2 * n$ values at level n.

Initialization:
$A[*, *] = \infty$
$A[p_1, O] = ||O + p_1||$, where O is the origin

For i = 1 to n-1:
    For j = 0 to i-1:
        $A[i+1, j] = A[j, i+1] = min(A[j, i+1], A[i+1, j], A[i, j] + ||p_i + p_{i+1}||)$
        $A[i+1, i] = A[i, i+1] = min(A[i, i+1], A[i+1, i], A[i, j] + ||p_j + p_{i+1}||)$

The minimum-length path will be in row $A[*, n]$.