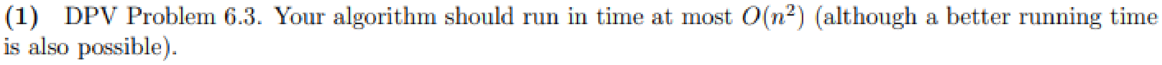
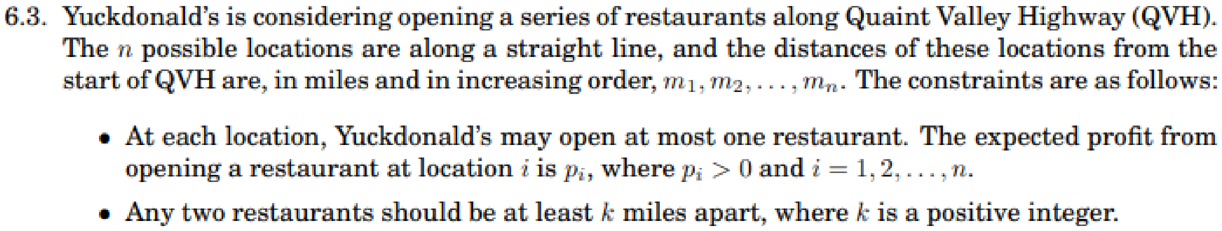


(1) DPV Problem 6.3. Your algorithm should run in time at most $O(n^2)$ (although a better running time is also possible).



6.3. Yuckdonald's is considering opening a series of restaurants along Quaint Valley Highway (QVH). The n possible locations are along a straight line, and the distances of these locations from the start of QVH are, in miles and in increasing order, $m_1, m_2, \ldots, m_n$. The constraints are as follows:

- At each location, Yuckdonald's may open at most one restaurant. The expected profit from opening a restaurant at location i is p_i, where $p_i > 0$ and $i = 1, 2, \ldots, n$.
- Any two restaurants should be at least k miles apart, where k is a positive integer.

i = # of possible locations (sorted based on distance)
j = List of distances from QVH (m(j) is the distance between QVH and the i location)
p = list of expected profits (p(i) is the profit from the restaurant at i location)
k = min distances between pair of restaurants
dp(#i) = list of max profits based on each location

dp(i) = max[dp(i-1), p(i) + dp(j)]

Where the max profit is dp(n)
Runs in O(n log n) sorting and binary search