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.

Give an efficient algorithm to compute the maximum expected total profit subject to the given constraints.

Scratch

M = 100, 200, 300, 400, 500, 600, 700

P = 500, 600, 300, 700, 800, 400, 900

k = 150

Problem Formulation

T[i] = maximum total profit in opening restaurants (following given constraints) from the available locations m_1, m_2, ……m_i, where m_i is included.

Recurrence

T[i] = if $m_i > k$,

Look for all the milestones m_t such that m_t is at least k miles apart from m_i, then find max of T[t] and sum with P[i], if no such milestones take P[i]

If $m_i <= k$,

P[i]

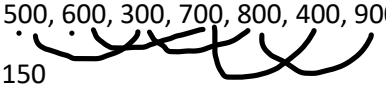
Base case

For all mi <= k, T[i] = P[i]

Sample Run (rhymes with Temple Run)

M = 100, 200, 300, 400, 500, 600, 700

P = 500, 600, 300, 700, 800, 400, 900



k = 150

T = 500, 600, 500+300, 600+700, 500+300+800, 600+700+400, 500+300+800+900

Return Value

Max of T

PseudoCode

```
for i = 1-> n        ---------- (1)
        if (M[i] <= k)
                T[i] = P[i]
        Else
                maxt = 0
                for t = i-1 -> 1    ----------  (2)
                        if ((M[i] – M[t]) >= k)
                                if (T[t] > maxt)
                                        maxt = T[t]
                T[i] = maxt + P[i]
Return max(T)   ----------(3)
```

Runtime Complexity

The Loops 1 and 2 are nested, hence will take O(n^2) time in worst case and Line 3 max operation will take O(n).
Hence, total runtime complexity is O(n^2).