

**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 m1, m2, ……mi, where mi is included.

**Recurrence**

T[i] = if mi > k,

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

If mi <= 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).