Assignment – 6

Answer to the Question No. - 1

(a) We need to examine every possible subset of the n potential locations and check if each subset satisfies the constraint. For each subset that satisfies the distance constraint, we compute the total revenue and keep track of the maximum revenue obtained. Each location can either be included or excluded from a subset, so there are 2^n possible subsets of n locations.

If we store the location in increasing order, we can verify the distance constraint within a subset by simply checking consecutive elements. This check can be done in O(n) times for each subset, as we only need to go through each location in the subset once to ensure the 10 mile requirement.

So the total complexity in brute is $O(n.2^n)$.

(b) Recursive definition for a[j]:

Base case: a[0] = 0

Otherwise: a[j] = max (a[l(j)] + t[j], a[j-1])

Where a[j-1] represents the optimal solution without including the toll booth at x_j and a[l [j]] + t[j] represents the total revenue if we include the toll booth at x_j along with the optimal solution for locations up to l(j).

(c)
```
Toll_DP (a, t, l){
        a[0] = 0;
        for i = 1 to n {
                if (a[l[i]] + t[i] > a[i-1]) a[i] = a[l[i]] + t[i];
                else a[i] = a[i-1];
        }
}
computeL (a, l){
        l[0] = 0;
        for i = 1 to n { for j = i-1 to 1 { if (a[i] – a[j] >= 10) {
                l[i] = j; break;
}}}}
```

(d) The algorithm makes one call of the procedure compute(a, l) and another call to Toll_DP(a, t, l) where they take $O(n^2)$ and O(n) time, respectively. Hence, overall running time for the algorithm is $O(n^2)$.