Homework 6

Shadman Latif (quk219)
Ubaidullah Khan (del028)

Q2)

a) Given set of locations $\{x_1, x_2, \ldots \ldots x_n\}$
For each of these locations, we have 2 possible outcomes. They are either part of the solution or not. Therefore, for 'n' number of locations, we will get a set of possible solutions of size 2^n.

b) Considering the example above,
$a[j-1] = 60$; $t_j = 50$; $l(j) = 10$
$a[l(j)] = 50$

Now, we have to choose the optimal amount of toll between our previous optimal amount & our new optimal amount up to $l(j) + t$ (of the current location)
This is with regards to the fact that $l(j)$ is updated in each step

$$\begin{cases} a[0] = 0 & \longrightarrow \text{Base case} \\ a[j] = \max((a[l(j)] + t_j), a(j-1]) \end{cases}$$

c)

```
A = sol [n]
Toll - optimization ( x, t)
{
   A [0] = 0

     for i = 1 to n
      {
        int l(j) = get_lj (i)

             A [i] = max (a [l(j) + t[i] , a [i - 1]])

             }

  }
  int get_lj ( x, index)
   {
      int lj = 0
       for (i = index - 1 to 1)
        {
```

Type your text

```
        if (x[index] - x[i] >= 10)
        { Lj = i; break; }

        return Lj
        }
  }
```

d) According to the code, there are 2 loops.

An outer loop for filling in the 'A' array that hold the optimal values.

An Inner loop to find Lj.

Therefore, The run time is $O(n^2)$.