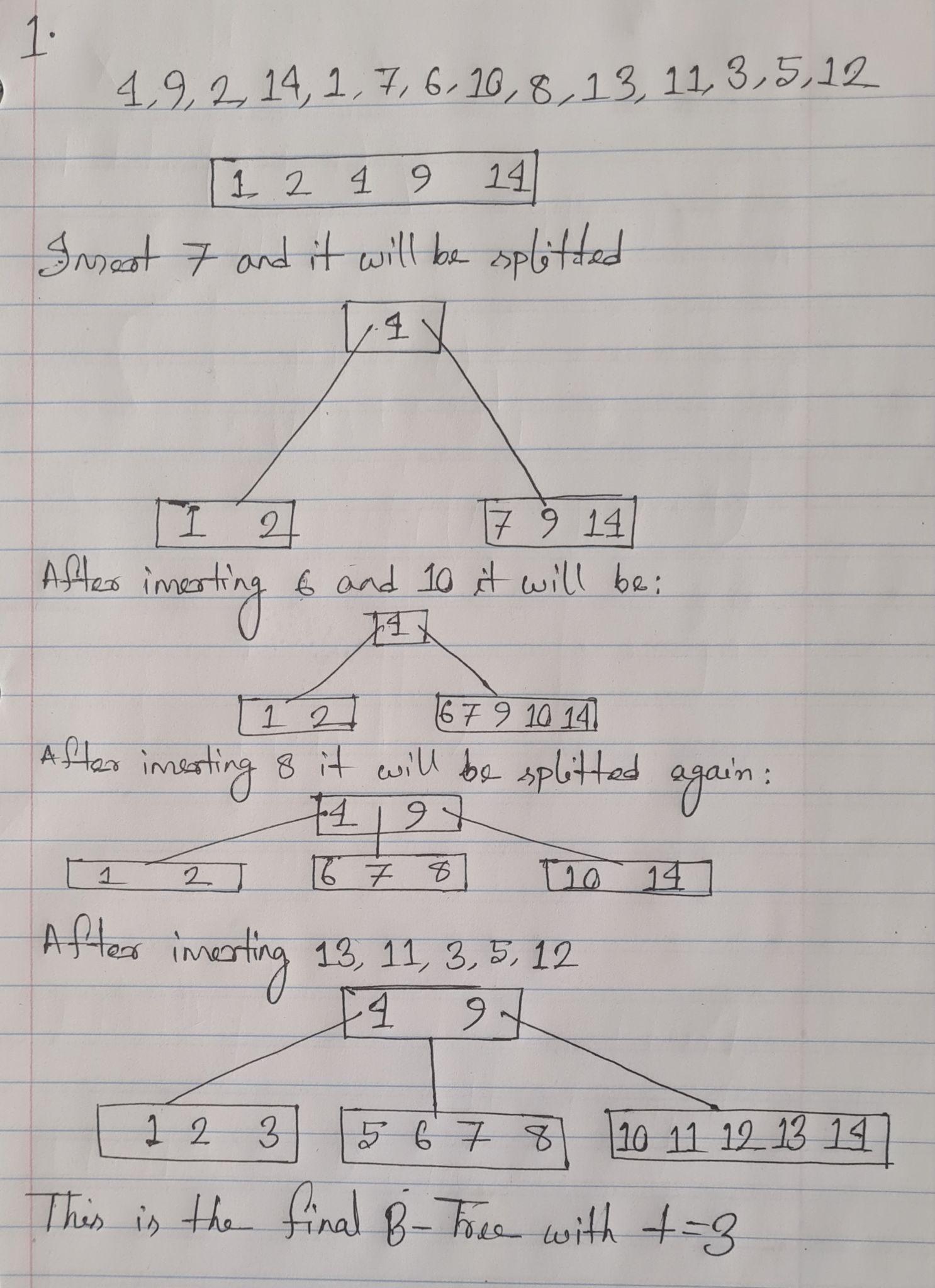
**CS 5633: Analysis of Algorithms**

## **Homework 6**

1. Here, t = 3. So, each node has at most 6 children and can store at most 5 values.



2.a)

Here, the possible toll booth positions = 2n

We will need to calculate the amount of money that can be earned for which required time is O(n) for all the possible positions. So, overall running time = O(2nn)

b) For this, we will need to find the index of the rightmost toll position left to j-th position which is at least 10 miles apart from j-th position. Let the index of that position be x.

Recursive definition of a[j]:

if(j == 1)

return a[1] //Base case

else

return max(a[j-1], t[j]+a[x]) [definition of x is provided at the first line of 2.b]

c)

find\_index(a, l):

for i = 1 to n:

j = i -1

while(j != 0):

if a[i] - a[j] >= 10:

l[i] = j

break

j--

find\_max\_DP(a, t, l):

a[1] = a[1]

for j = 2 to n:

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

d) Here, find\_index has a nested loop. So, the running time = O(n2) and find\_max\_DP has one for loop. So, running time =O(n). So, overall running time will be O(n2).