Design and Analysis of Algorithm Lab4

Name:V.Prasanna Vyshnavi

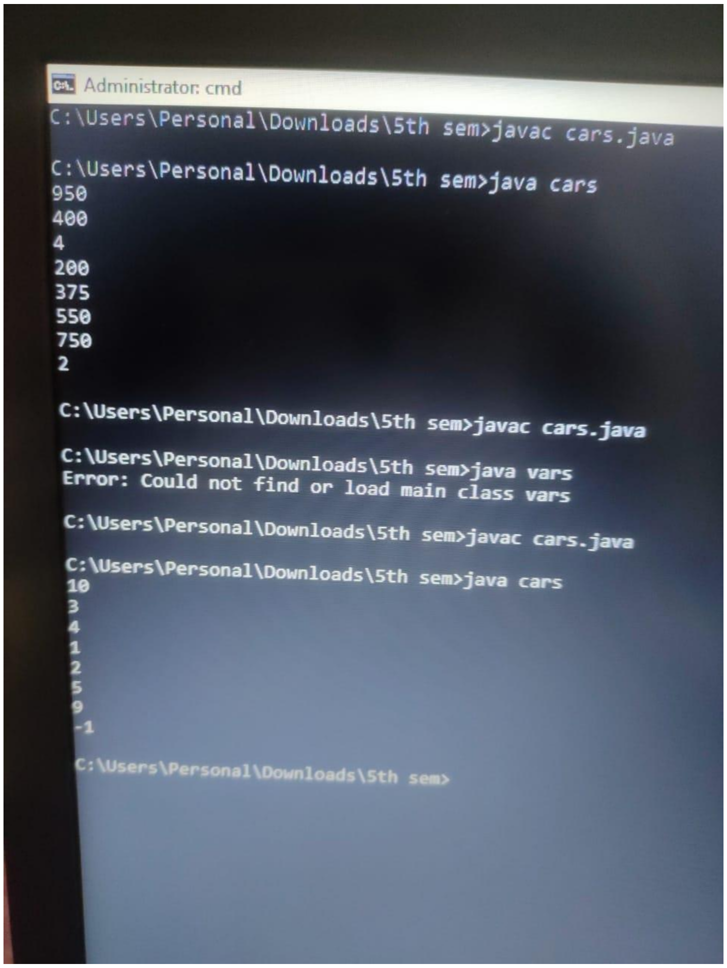
Regno:19BCE7661

Car fueling problem:

Code:

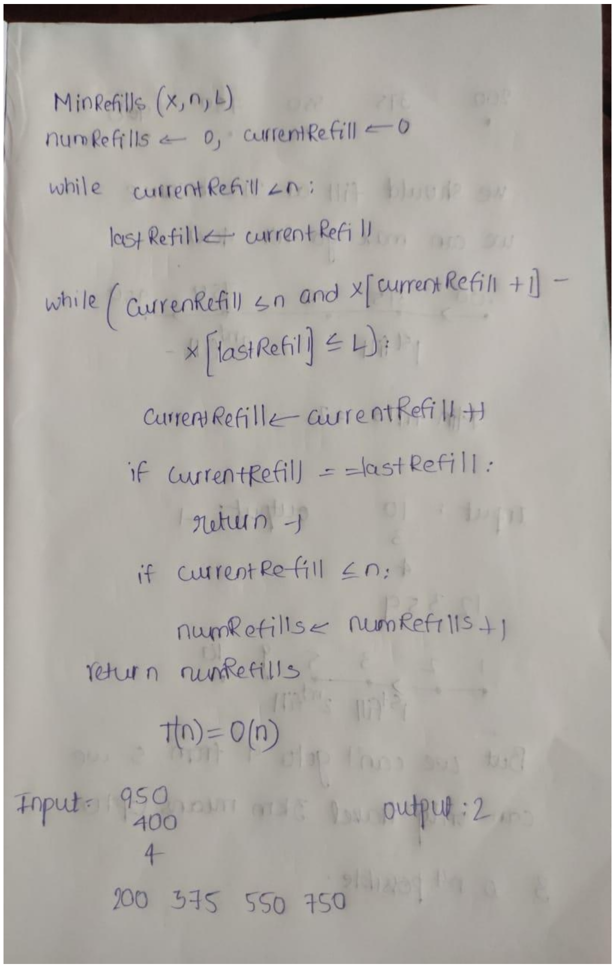
```
import java.util.*;
import java.io.*;
public class cars {
static int compute_refills(int dist,int tank,int stops[],int n){

int current_refills=0;
int num_refills=0;
int last_refill=0;
while(current_refills<=n) {
last_refill = current_refills;
while ((current_refills <= n) && (stops[current_refills + 1] - stops[last_refill])
<= tank) {
current_refills = current_refills + 1;
}
if (current_refills == last_refill)
return -1;
if (current_refills <= n)
num_refills = num_refills + 1+1;
}
return num_refills;
}
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
int dist = scanner.nextInt();
int tank = scanner.nextInt();
int n = scanner.nextInt();
int stops[] = new int[n*n*n];
for (int i = 0; i < n; i++) {
stops[i] = scanner.nextInt();
}
System.out.println(compute_refills(dist,tank,stops,n));
}
}
```

Output:



```
C:\Users\Personal\Downloads\5th sem>javac cars.java

C:\Users\Personal\Downloads\5th sem>java cars
950
400
4
200
375
550
750
2

C:\Users\Personal\Downloads\5th sem>javac cars.java

C:\Users\Personal\Downloads\5th sem>java vars
Error: Could not find or load main class vars

C:\Users\Personal\Downloads\5th sem>javac cars.java

C:\Users\Personal\Downloads\5th sem>java cars
10
3
4
1
2
5
9
-1

C:\Users\Personal\Downloads\5th sem>
```

Asymptotic Analysis:

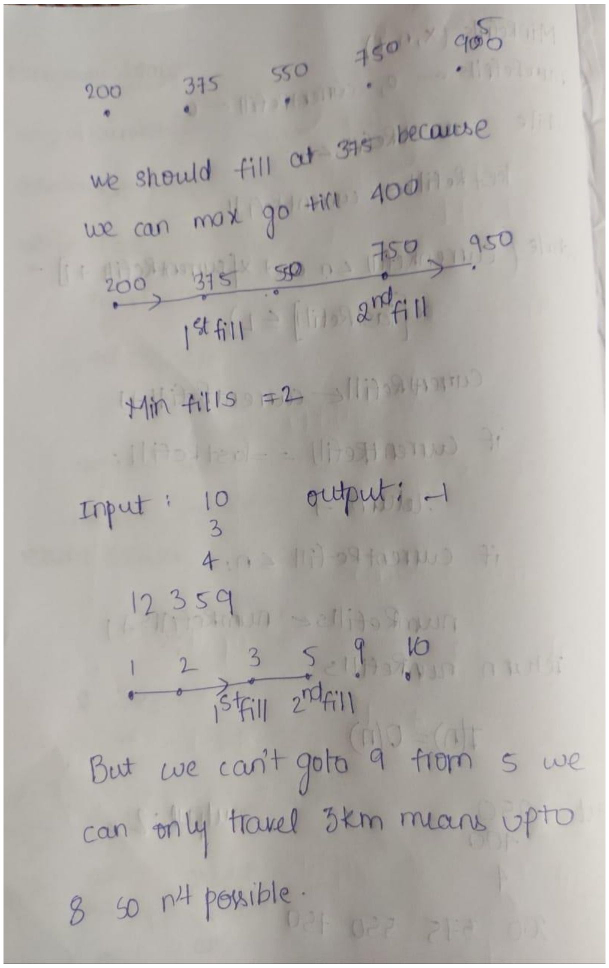


```
MinRefills (x, n, L)
numRefills ← 0, currentRefill ← 0
while currentRefill < n :
      last Refill ← current Refill
while ( CurrenRefill ≤ n and x[currentRefill + 1] -
            x[lastRefill] ≤ L ):
          CurrentRefill ← currentRefill +1
        if currentRefill = = lastRefill :
              return -1
         if currentRefill ≤ n :
              numRefills ← numRefills + 1
    return numRefills
```

$T(n) = O(n)$

Input : 950
400
4
200 375 550 750

output : 2



200 375 550 750 950

we should fill at 375 because we can max go till 400

200 375 550 750 950

1st fill 2nd fill

Min fills = 2

Input : 10 output: -1
3
4
12359

1 2 3 5 9 10

1st fill 2nd fill

But we can't goto 9 from 5 we can only travel 3km means upto 8 so n4 possible.

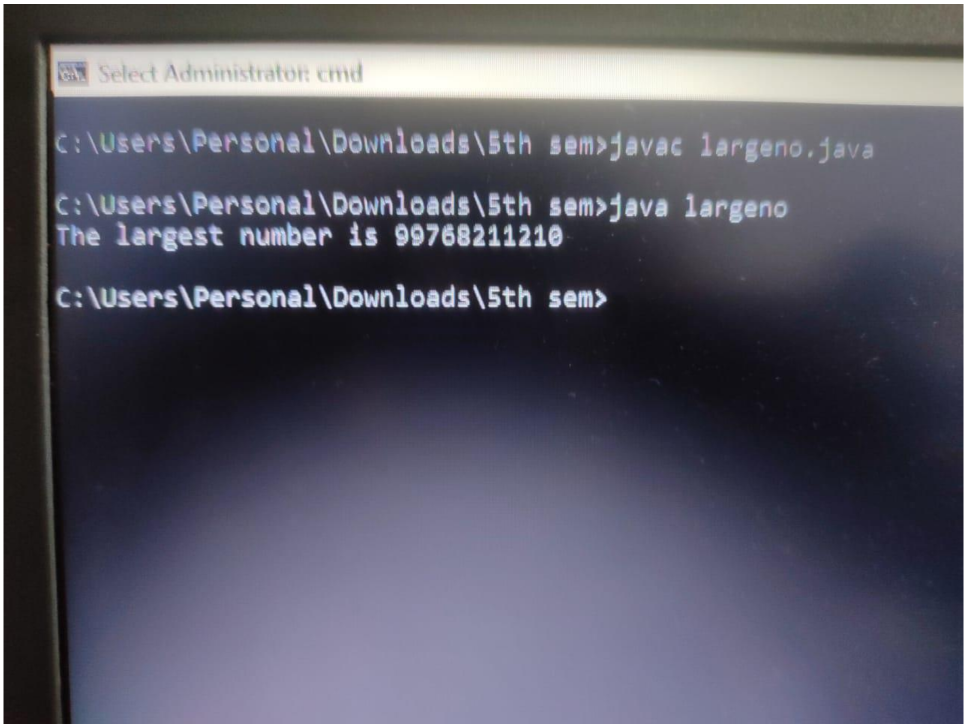
Maximum salary:

Code:

```
import java.util.*;

import java.util.Arrays;

import java.util.Collections;

import java.util.List;

import java.util.stream.Collectors;


public class largeno
```

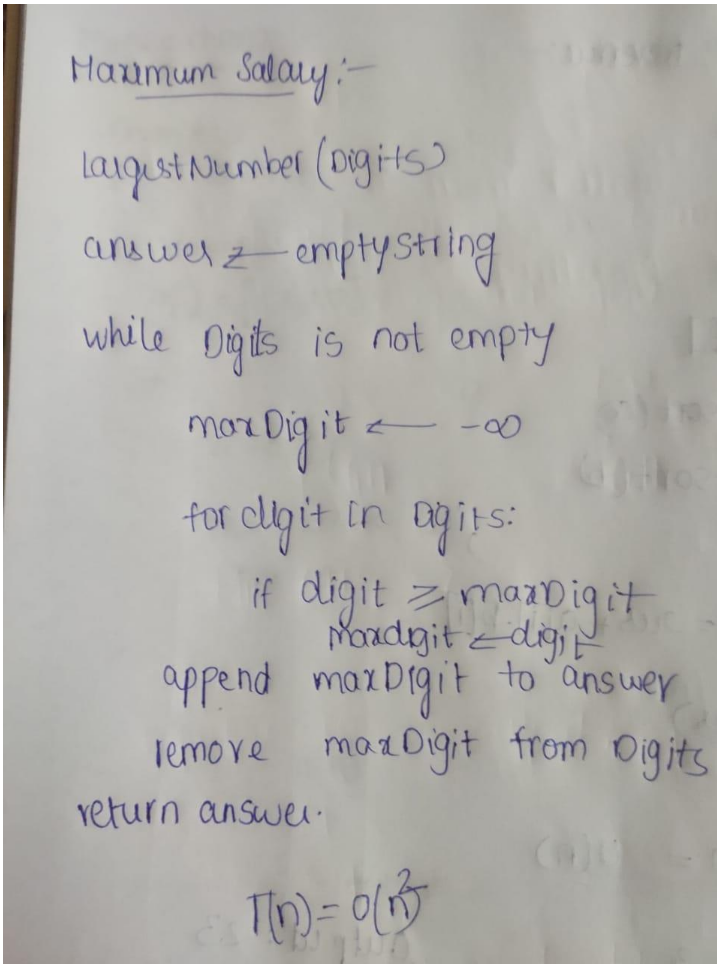
```
{
public static String findLargestNumber(List<Integer> nums)
{
Collections.sort(nums, (a, b) -> (String.valueOf(b) +
a).compareTo(String.valueOf(a) + b));
return nums.stream()
.map(Object::toString)
.collect(Collectors.joining(""));
}

public static void main(String[] args)
{
List<Integer> numbers = Arrays.asList(10, 68, 97, 9, 21, 12);

String largestNumber = findLargestNumber(numbers);
System.out.println("The largest number is " + largestNumber);
}
}
```

Output:



```
C:\Users\Personal\Downloads\5th sem>javac largeno.java

C:\Users\Personal\Downloads\5th sem>java largeno
The largest number is 99768211210

C:\Users\Personal\Downloads\5th sem>
```

Analysis:



Maximum Salary:-

```
LargestNumber(Digits)
answer $\leftarrow$ emptystring
while Digits is not empty
        maxDigit $\leftarrow -\infty$
        for digit in Digits:
                if digit $\geq$ maxDigit
                        maxdigit $\leftarrow$ digit
        append maxDigit to answer
        remove maxDigit from Digits
return answer.
```

$T(n) = O(n^2)$