

DAA - Lab3

Feb 8

Vishal Gauba
1410110501

Problem Statement

Suppose you were to drive from point A to point B. Your gas tank with a capacity C , when full, holds enough gas to travel m miles. In the case that the rate at which you can fill your tank at a gas station is r (in liters/minute), so if you stop to fill your tank from 2 liters to 8 liters, you would have to stop for $6/r$ minutes. Give the most efficient greedy solution, where you need to minimize the total time you stop for gas filling?

Input

1. Distance between A and B
2. Max distance between any two petrol stations
3. Number of petrol stations
4. Capacity of the car's petrol tank
5. Time to fill per litre of petrol

Output

1. Time to fill petrol each time
2. Total time taken
3. Total distance travelled

Algorithm

1. Generate n random numbers between 0 to 100
2. Sort them
3. Check if the total sum is 100
4. Check if the maximum difference between any two is 20
5. If yes, select.
6. If no, discard and go to step 1

7. Check if you can reach the next station with current petrol
8. If not, fill the petrol only as much required to reach the next station

Source Code

```
#include <stdio.h>
#include "stdlib.h"
#include "time.h"

int max_capacity=10;
int max_diff=20;

int* generate_stations(int m, int n, int c, int ab_dist)
{
    srand(time(NULL));

    int remaining = ab_dist;
    int station_no = 1;
    int stations[8];

    stations[0]=0;

    while(station_no < 7){
        int Min = remaining/(n-station_no);
        int Max = 20;

        int diff = Max-Min;
        int d = (int) (((double)(diff+1)/RAND_MAX) * rand() + Min);

        // int d = (int)(((randmax-randmin+1)*((double)rand()/RAND_MAX)) + randmin);
        // printf("%d\n",Min);
        stations[station_no] = d;
        station_no++;
        remaining -= d;
    }

    int i;
    for (i = 0; i < 8; ++i)
    {
        // printf("%d\n",stations[i]);
    }
    return stations;
}

int main(){

    /* test case */
    int station_list[]={0,20,11,18,8,7,15,12,9};
    /* variables */
    int ab_dist = 100;
    int m = 20;
```

```

int n = 8;
int c = 10;

/* station generation algorithm*/
int* staton_list = generate_stations(ab_dist,m,n,c);

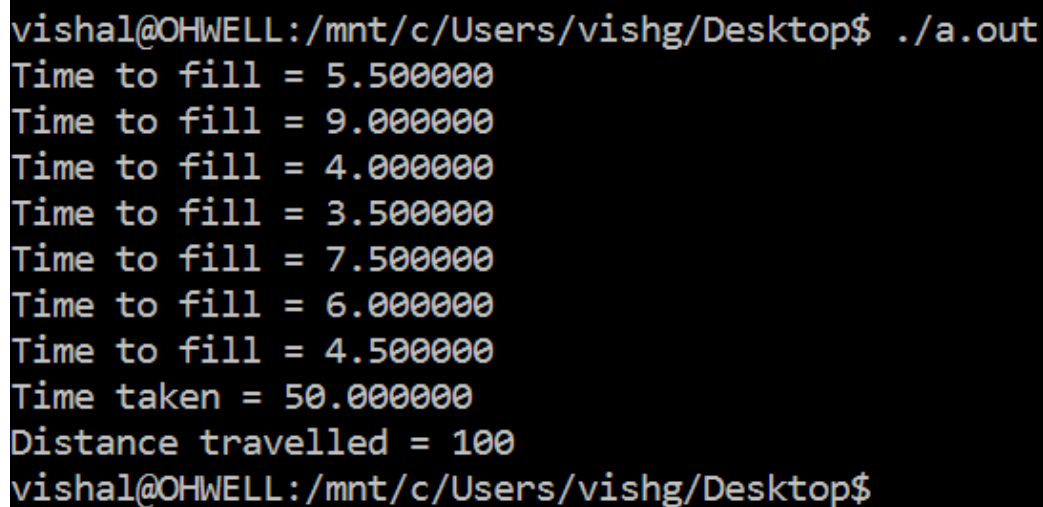
int dist_travelled=0;
int station_id=1;
float present_petrol=10.0;
float petrol_fill=10.0;
float a,b; int sum;
while(dist_travelled<100){
    if(present_petrol*2<(float)(station_list[station_id])){

        a = (float)(station_list[station_id] - (present_petrol*2));
        printf("Time to fill = %f \n",(float)a/2);
        petrol_fill += (float)a/2;
    }
    present_petrol=0;
    if(dist_travelled<100){
        dist_travelled=dist_travelled+station_list[station_id];
    }
    station_id++;
}

printf("Time taken = %f \n",petrol_fill);
printf("Distance travelled = %d \n",dist_travelled);
return 0;
}

```

Screenshots of solution



```

vishal@OHWELL:/mnt/c/Users/vishg/Desktop$ ./a.out
Time to fill = 5.500000
Time to fill = 9.000000
Time to fill = 4.000000
Time to fill = 3.500000
Time to fill = 7.500000
Time to fill = 6.000000
Time to fill = 4.500000
Time taken = 50.000000
Distance travelled = 100
vishal@OHWELL:/mnt/c/Users/vishg/Desktop$

```