DSA - Mini Project Report

Group 57

Contents

1 The Team

S. No.	Name	Roll Number
1	Aditya Harikrish	2020111009
2	Akshit Gureja	2020112004
3	Aishwaryabharathi Upadhyayula	2020102060
4	Jaishnav Yarramaneni	2020102059
5	Rishabh Srivastava	2020101047

2 Data Structures Used

1. Arrays
2. Vectors (user-defined), modelled after std::vectors in C++.

```
    struct vector {
        size_t size, capacity;
        int* arr;
    };

```

- arr is an array which stores integers.
- size is the number of elements currently stored in the array arr
- capacity is the space allocated to arr

malloc and realloc from stdlib.h are used to allocate/ resize the space allocated to arr when necessary; and each resizing allocated twice the previous capacity to arr .

3 Algorithms Used and Their Complexity

3.1 Time complexity of the functions

Function Name/ Purpose	Time Complexity
UpdatePeople	$O(1)$ if $size < capacity$ $O(size)$ if $size = capacity$
getPrimaryContacts	$O(X^2 * K *$stationsVisited by the person)
getPrimaryContacts_print	$O(X^2 * K *$ stationsVisited by the person)
getStationContacts_primary_print	$O(X^2 * K *$ stationsVisited by the person)
getSecondaryContacts	$O(X^2 * stationsVisitedbyprimarycontacts$
Initialise a vector	$O(1)$
Pushback into a vector	$O(1)$ if $size < capacity$ $O(size)$ if $size = capacity$
Popback from a vector	$size < capacity$ $O(size)$ if $size = capacity/2$
Delete a vector	$O(1)$
move_forward_one_day	$O(NK)$
Get the top 3 safest and shortest paths	$O(n^3)$
Compare 2 paths	$O(n)$
Get the safest station not yet traversed	$O(n)$
Get the path from a source to destination in a vector	$O(n)$
Initialise a person	O(1)
Add travel	Same as pushback
Get status	$O(1)$
Initialise a station	$O(1)$
list_positive	$O(K)$
list_primary	$O(K)$
list_secondary	$O(K)$
list_positive_at_s	$O(K)$
list_primary_at_s	$O(K)$
list_secondary_at_s	$O(K)$
Get location of a person	$O(1)$

4 Division of Work

1. **Aditya:** Coded the basic input and output formats (UI), organised the program into multiple files, built the basic structs of persons and stations and coded the functions associated with them. Additionally, replicated vectors (dynamic auto-resizing arrays)

in C, wrote this report and contributed to README.

2. **Akshit:** Coded the second major task, which is to find the top three safest and shortest paths and added Python scripts to visualise data.

3. **Aishwaryabharathi:** Wrote the code to find out all the secondary contacts in the first major task.

4. **Jaishnav:** Wrote the code to find out all the primary contacts in the first major tasks, wrote most of the README.

5. **Rishabh:** Coded the third major task, which is to handle queries like

   (a) the status of a person

   (b) the location of a person

   (c) the list of all COVID-positive people, primary contacts and secondary contacts at a particular station

   (d) the list of all COVID-positive people, primary contacts and secondary contacts across all stations