

Patient Manager

Leon Kosarev

Version: 16.08.20

General Description:

In light of recent events, I've decided to build a program that will help manage information about the spread of a virus. The program will make use of the C++ standard template library (STL), data structures and other tools, in order to implement the needed functionality under optimal complexity.

Information Storage:

The program will store patient information by assigning each patient with a unique ID number according to chronological order. In other words, the N-th patient will receive the ID number N. In addition, the program will keep track of the number of people infected by each patient (infection count).

Current Functionality:

Init(string zeroPatientName, int zeroPatientAge, Time registrationTime) - Initializes a patient database containing information about the first known patient. - **$O(1)$**

AddPatient(int spreaderID, string name, int age, Time registrationTime) - Adds a new patient to the system (infected by patient with ID number - spreaderID). - **$O(1)$** (amortized)

PrintSuperSpreaders(int K) - Prints the ID numbers of K patients with the highest infection count. - **$O(K)$**

PrintInfectedBy(int patientID) - Prints the ID numbers of all K patients, directly or indirectly infected by patient with a given ID. - **$O(K)$**

AddToRiskGroup(int patientID) - Indicate that an existing patient is in a risk group. - **$O(\log n)$**

RiskGroupInTimeRange(Time start, Time end) - Prints the number of risk group patients, registered within a certain time range. - **$O(\log n)$**

Implementation:

The following section will contain a general summary of the implementation, as well as the data structures used to maintain the complexities mentioned above.

Data Structures:

- Directed graph, implemented using adjacency lists. An edge (x,y) will exist in the graph if and only if patient y was infected by patient x.
- Rank List – A list, where each node represents a group of patients with the same infection count and contains a list of ID numbers of those patients.

- Dynamic array – Contains the information of all patients in the system.
- SumTree – A balanced BST, where each node contains an integer, indicating the quantity of the key representing the node.

