

Advanced Programming

Sino-Italian Campus, Tongji University (Shanghai)

This program manages rain precipitation records collected by rain stations in a region in a certain year.

addStation(char* name, double rainQuantity)

sets the quantity of rain collected by the station *name*

nearStations(char* name, double interval)

returns the quantity of rain collected by station name, and also names and rain quantity of all stations with a rain quantity within *interval*

The time complexity of nearStations() should be the best possible, and at least less than $O(n)$

Example

```
...
addStation(Station_J 120.23);
addStation(Station_B 150.13);
addStation(Station_Z 1988.90);
addStation(Station_A 149.3);
nearStations(Station_B, 5.0);
    Station_B: amount of rain 150.13
    Similar stations (range: 5)
    Station_A: amount of rain 149.3
...
```

1. What information should we record?

A pair of data:

- station name
- rain

2. What operations the program needs to perform

nearStations

- retrieve information about the given station
- print the station name and rain value
- for each station
 - o if it is within the interval, print the station name and rain value

assumption: there are M stations within the interval with $M \ll n$

where n is the total number of stations

3. What data structures are going to use?

Different options:

O1: Array of station information (would require assumption about max number of stations)

- retrieve is $O(n)$
- find stations within range is $O(n)$

O2: Unbounded array (does not require any assumption about max number)

same complexity as bounded array

O3: Sorted (unbounded) array by station name:

- retrieve is $O(\log n)$
- find stations within range is $O(n)$

Overall $O(n)$

O4: Sorted (unbounded) array by rain:

- retrieve station is $O(n)$
- find stations within range is $O(M)$

Overall $O(n)$

O5: Two ordered arrays: N sorted by name, R sorted by rain value

- retrieve station is $O(\log n)$ (binary search in array N)
- find stations within range is $O(M)$

Overall is $O(\log n)$

O6: Hash table to retrieve station info, and array sorted by rain value, provided that in the hash table you store the index of the info in the sorted array

- retrieve station is $O(1)$
- find stations within range is $O(M)$

Overall $O(1)$

O7: Hash table to retrieve station info and a BST sorted by rain value

- retrieve station $O(1)$
- find stations within range is $O(\log n)$

Overall $O(\log n)$

4. implement it in C