Standard Data Structures: Sets and Maps

ICT and SCT Captains

October 2020

1 Introduction

Most programming languages come with built-in data structures that are useful in competitive programming. Of those data structures, this lecture will go over sets and maps.

2 Sets

A set is a data structure that allows quick random access, addition, and removal of values, but can't contain repeats of items.

2.1 Unordered Sets

Unordered Sets(HashSet in java) use hashing to allow very quick access to elements. The downside is that we don't know the order of the elements in the set.

Effciencies:

- Adding a value: $O(1)$
- Removing a value: $O(1)$
- Checking if the set contains a value: $O(1)$
- Looping through the set: $O(N)$

2.2 Ordered Sets

Ordered Sets(TreeSet in java) use a binary search tree to allow quick access to elements. Values in the set are kept in sorted order.

Effciencies:

- Adding a value: $O(log(N))$
- Removing a value: $O(log(N))$

- Checking if the set contains a value: $O(log(N))$
- Finding the smallest or largest value: $O(log(N))$
- Finding the smallest value bigger than a given value: $O(log(N))$
- Finding the largest value smaller than a given value: $O(log(N))$
- Looping through the set: $O(N)$

3 Maps

Maps are like set, but instead of storing values, we store keys. Each key "maps", or is linked to a value, and we must access values using their keys. Like in sets, keys can't repeat. However, the values the keys map to can.

3.1 Unordered Maps

Unordered Maps(HashMaps in java) are pretty much just the map version of an unordered set.

Effciencies:

- Adding a key-value pair: $O(1)$
- Removing a key-value pair: $O(1)$
- Updating a key-value pair: $O(1)$
- Checking if the map contains a key: $O(1)$
- Checking if the map contains a value: $O(N)$
- Getting the value for a certain key; $O(1)$
- Looping through the map: $O(N)$

3.2 Ordered Sets

Ordered Maps(TreeMaps in java) are like their set counterparts. Keys are kept in sorted order.

Effciencies:

- Adding a key-value pair: $O(log(N))$
- Removing a key-value pair: $O(log(N))$
- Updating a key-value pair: $O(log(N))$
- Checking if the map contains a key: $O(log(N))$

- Checking if the map contains a value: $O(N)$
- Getting the value for a certain key; $O(1)$
- Finding the smallest or largest key: $O(log(N))$
- Finding the smallest key bigger than a given key: $O(log(N))$
- Finding the largest key smaller than a given key: $O(log(N))$
- Looping through the map: $O(N)$

4 Multisets and Multimaps

If you code in C++, there are versions of set and map that allow you to repeat values(Multiset and Multimap respectively). In java, there is no such data structure, but we can emulate a multiset using a map where the key stores the value and the value the key maps to store the number of times it occurs.

5 Practice Problems

- USACO 2015 January Contest, Silver: Stampede
- USACO 2015 December Contest, Platinum: High Card Low Card