

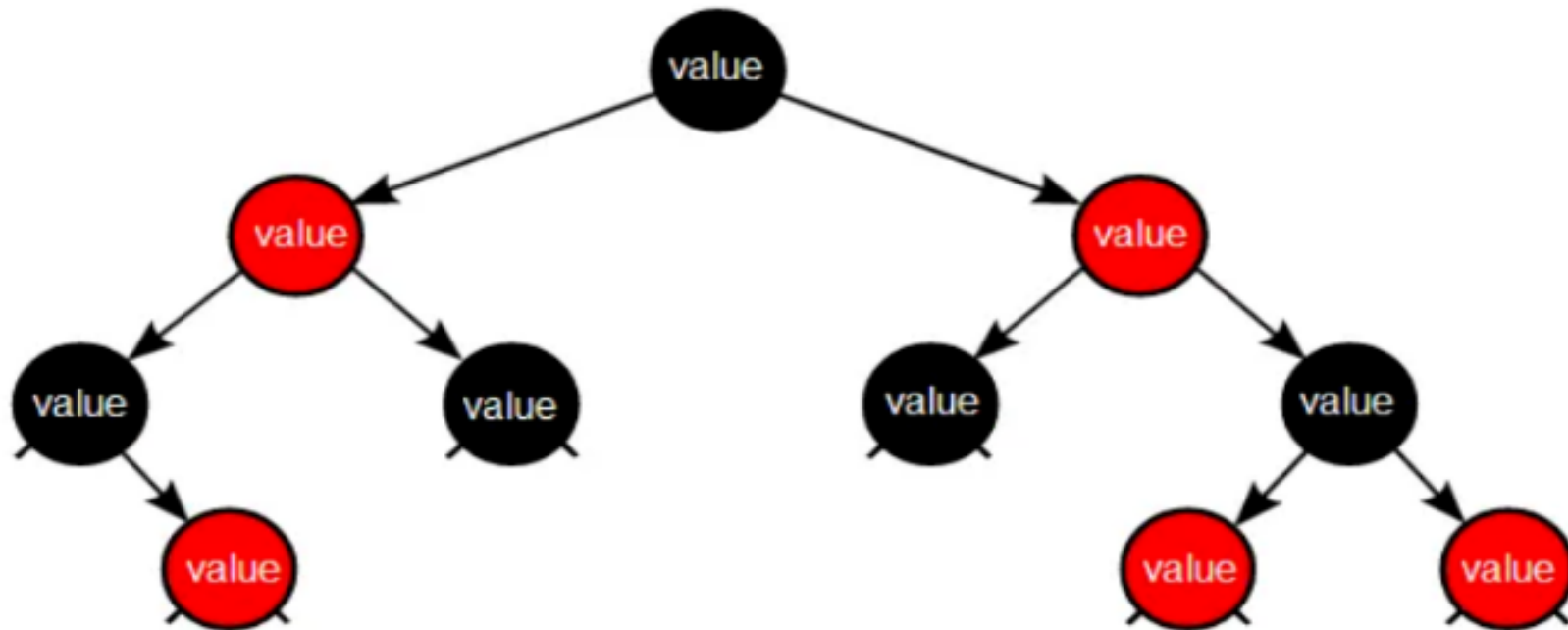


Set Trong C++

SET, MULTISSET, UNORDERED_SET

Set là gì?

1. Set là một container cực kì mạnh mẽ của ngôn ngữ lập trình C++, sử dụng thành thạo Set là một kỹ năng cơ bản mà bạn cần đạt được. Set sẽ giúp code của các bạn trở nên tối ưu và ngắn gọn hơn rất nhiều.
2. Set có tốc độ tìm kiếm phần tử cực kì nhanh.



1

Set.



Set

Tính Chất:

Set là một container mà mỗi phần tử trong đó là duy nhất, tức là sẽ không có 2 phần tử có giá trị giống nhau cùng tồn tại trong set.

Các phần tử trong set được sắp xếp theo thứ tự tăng dần về giá trị số và tăng dần về thứ tự từ điển nếu là chuỗi ký tự.

Khai Báo:

```
set <data_type> set_name;
```



Một số hàm thông dụng trong Set:

- Hàm **insert** : thêm một phần tử trong set.
- Hàm **size**: trả về số lượng phần tử trong set.
- Hàm **erase**: Xóa 1 phần tử khỏi set với độ phức tạp là $O(\log n)$.
- Hàm **clear**: xóa mọi phần tử trong set.
- Hàm **empty**: kiểm tra set rỗng, nếu rỗng trả về true, ngược lại trả về false.
- Hàm **find** : kiểm tra sự tồn tại của một phần tử nào đó trong set, đây là một hàm được sử dụng rất nhiều của set vì độ phức tạp của nó là $O(\log n)$.
- Hàm **count** : Hàm này dùng để đếm số lần xuất hiện của 1 phần tử trong set, đối với set thì 1 phần tử sẽ xuất hiện 1 hoặc 0 lần. Độ phức tạp $O(\log n)$.



2

MULTISET

MULTISET

-Tính Chất:

Multiset tương tự như set nhưng có thể lưu nhiều phần tử có giá trị giống nhau, các phần tử này cũng được sắp xếp theo thứ tự tăng dần. Các hàm của multiset giống y hệt các hàm của set chỉ khác một chút ở hàm find, count, erase.

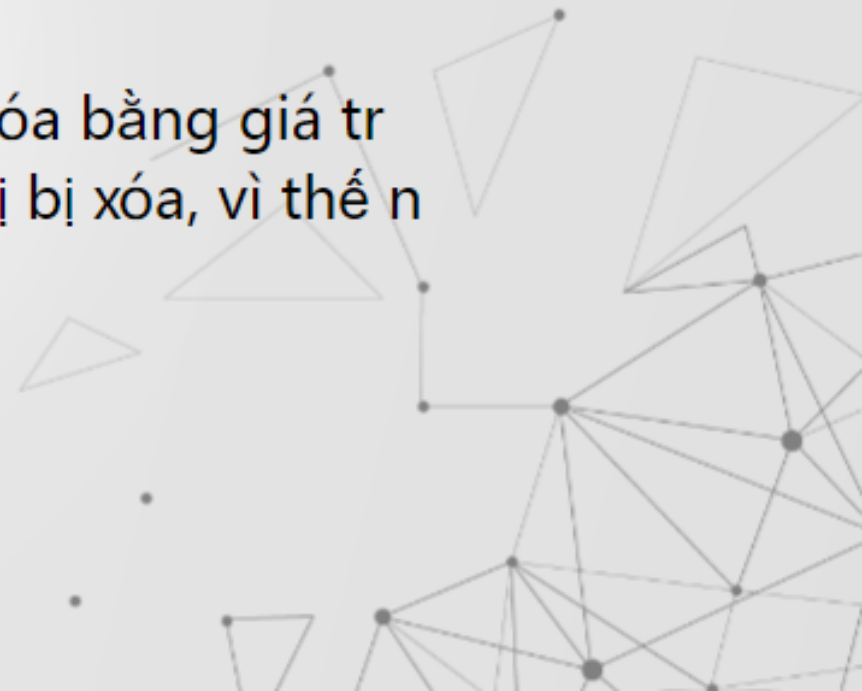
-Khai Báo:

```
multiset <data_type> multisetset_name;
```



MỘT SỐ HÀM TRONG MULTISSET

- Hàm **find** : Vì multiset có thể chứa nhiều phần tử giống nhau nên hàm find sẽ trả về iterator đến vị trí đầu tiên của phần tử có giá trị cần tìm kiếm.
- Hàm **count** : Hàm này dùng để đếm số lần xuất hiện của 1 phần tử trong multiset. Nếu phần tử không có trong multiset sẽ trả về 0. Độ phức tạp $O(\log n)$.
- Hàm **erase** : Khi sử dụng hàm erase nếu bạn xóa bằng giá trị i multiset sẽ xóa hết mọi phần tử có cùng giá trị bị xóa, vì thế nếu muốn 1 phần tử bạn phải xóa bằng iterator



3

UNORDERED_SET.



UNORDERED_SET

-Tính Chất:

Unordered_set khác với set và multiset ở tốc độ của các hàm phổ biến như count, find, và erase. Còn cách sử dụng thì không có gì khác biệt so với set. Ở set và multiset các hàm có độ phức tạp là $O(\log N)$.

Ở unordered_set tốc độ của các hàm trong trường hợp tốt nhất là $O(1)$ còn tệ nhất có thể lên đến $O(N)$.

-Khai Báo:

```
unordered_set <data_type> unordered_set_name;
```



THANKS

Full House Tất Cả Là Một Nhà!

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), and infographics & images by [Freepik](#).

Please keep this slide for attribution.