

SET

```
#include <set>
```

Set

Khái niệm

Set là một container cực kì mạnh mẽ trong thư viện STL 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.

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 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 xâu kí tự.

Set

CÚ PHÁP

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

SỬ DỤNG SET



Các bài toán liên quan tới việc xóa, thêm, tìm kiếm một phần tử nào đó được thực hiện đi thực hiện lại nhiều lần.



Các bài toán liên quan tới các giá trị khác nhau của mảng



Một số hàm trong set

Hàm insert: thêm một phần tử trong set.

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {1, 1, 2, 1, 3, 4, 5};
    for(int i = 0; i < 7; i++){
        se.insert(a[i]);
    }
}
```

● se = {1, 2, 3, 4, 5}

Một số hàm trong set

Hàm size: trả về số lượng phần tử trong set.

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {1, 1, 2, 1, 3, 4, 5};
    for(int i = 0; i < 7; i++){
        se.insert(a[i]);
    }
    cout << se.size() << endl;
}
```

OUTPUT: 5

Một số hàm trong set

Hàm empty: kiểm tra set rỗng, nếu rỗng trả về true, ngược lại trả về false.
Hàm clear: xóa mọi phần tử trong set.

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {1, 1, 2, 1, 3, 4, 5};
    for(int i = 0; i < 7; i++)
        se.insert(a[i]);
    cout << se.size() << endl;
    if(se.empty())
        cout << "Empty !\n";
    else
        cout << "Not empty !\n";
    se.clear(); // set rỗng
    if(se.empty())
        cout << "Empty !\n";
    else
        cout << "Not empty !\n";
}
```

OUTPUT: 5
Not empty !
Empty !

Một số hàm trong set

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 này trả về iterator tới phần tử nếu nó tìm thấy, ngược lại nó trả về iterator `end()` của set khi phần tử đó không tồn tại trong set

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {5, 5, 1, 2, 3, 4, 5};
    for(int i = 0; i < 7; i++){
        se.insert(a[i]);
    }
    auto it = se.find(3);
    if(it == se.end()){
        cout << "NOT FOUND\n";
    }
    else{
        cout << "FOUND\n";
    }
}
```

OUTPUT: FOUND

Một số hàm trong set

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, có thể sử dụng hàm này để thay cho hàm find. Độ phức tạp $O(\log N)$.

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {5, 5, 1, 2, 3, 4, 5};
    for(int i = 0; i < 7; i++)
        se.insert(a[i]);
    cout << se.count(1) << endl;
    cout << se.count(6) << endl;
    if(se.count(3) != 0)
        cout << "FOUND\n";
    else
        cout << "NOT FOUND\n";
}
```

OUTPUT: 1
0
FOUND

Một số hàm trong set

Hàm erase: Xóa 1 phần tử khỏi set với độ phức tạp là $O(\log N)$, trước khi sử dụng hàm erase hãy đảm bảo phần tử bạn cần xóa tồn tại trong set nếu không sẽ xảy ra lỗi runtime error.

Xóa bằng giá trị

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {5, 5, 1, 2, 3, 4, 5};
    for(int i = 0; i < 7; i++)
        se.insert(a[i]);
    for(int x : se)
        cout << x << ' ';
    cout << endl;
    se.erase(3);
    for(int x : se)
        cout << x << ' ';
}
```

OUTPUT: 1 2 3 4 5
1 2 4 5

Một số hàm trong set

Hàm erase: Xóa 1 phần tử khỏi set với độ phức tạp là $O(\log N)$, trước khi sử dụng hàm erase hãy đảm bảo phần tử bạn cần xóa tồn tại trong set nếu không sẽ xảy ra lỗi runtime error.

Xóa bằng iterator

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {5, 5, 1, 2, 3, 4, 5};
    for(int i = 0; i < 7; i++)
        se.insert(a[i]);
    for(int x : se)
        cout << x << ' ';
    cout << endl;
    auto it = se.find(3);
    if(it != se.end())
        se.erase(it);
    for(int x : se)
        cout << x << ' ';
}
```

OUTPUT: 1 2 3 4 5
1 2 4 5

Duyệt set

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    set<int> se;
    int a[7] = {1, 1, 2, 1, 3, 4, 5};
    for(int i = 0; i < 7; i++)
        se.insert(a[i]);
    //Cách 1: Dùng For each
    for(int x : se)
        cout << x << ' ';
    cout << endl;
    //Cách 2: Dùng Iterator
    for(set<int>::iterator it = se.begin(); it != se.end(); ++it)
        cout << *it << ' ';
    cout << endl;
    //Cách 3: Dùng Auto
    for(auto it = se.begin(); it != se.end(); ++it)
        cout << *it << ' ';
}
```

OUTPUT: 1 2 3 4 5
1 2 3 4 5
1 2 3 4 5