Cấu trúc dữ liệu tập hợp và ánh xạ (từ điển)

Cấu trúc dữ liệu tập hợp (set)

- Khái niệm
  - Các phần tử cùng kiểu dữ liệu (C++)
  - Không trùng nhau
- Ví dụ
  - Mảng a = {1, 2, 1, 1, 2}
  - Tập hợp s = {1, 2}

Cấu trúc dữ liệu tập hợp (set)

- Thứ tự các phần tử
  - Mặc định (trong C++): thứ tự tăng dần
  - Thứ tự giảm dần: set<int, std::greater> s;
  - Hoặc theo định nghĩa:

```
struct cmpStruct {
  bool operator() (int const & lhs, int const & rhs) const
  {
    return lhs > rhs;
  }
};

set<int, cmpStruct > myInverseSortedSet;
```

Cấu trúc dữ liệu tập hợp (set)

- Cài đặt: kiểm tra trùng khi chèn
  - Lớp set
    - Các phần tử được sắp thứ tự
    - Sử dụng cây đỏ đen
    - Thời gian truy cập một phần tử: O(log(*n*))
  - Lớp unordered_set
    - Các phần tử không được sắp
    - Được cài đặt bằng bảng băm
    - Thời gian truy cập O(1)

Cấu trúc dữ liệu tập hợp (set)

- Cài đặt:
  - Một số phương thức
    - insert, erase, swap, clear, size, find
  - Duyệt phần tử

```
set<int>::iterator it;
for (it = s.begin(); it != s.end(); ++it)
    cout << ' ' << *it;
```

Cấu trúc dữ liệu tập hợp (set)

- Ứng dụng
  - Bài toán liên quan tới tập hợp
  - Ví dụ 1: Cho dãy *n* phần tử. Hãy đếm số phần tử khác nhau trong mảng.

```
set<int> s;
for (int i=0; i < n; i++)
    s.insert(a[i]);
return s.size();
```

Cấu trúc dữ liệu tập hợp (set)

- Ứng dụng
  - Ví dụ 2: Một vector *v* được gọi là vector beautifull nếu một số trong *v* chỉ xuất hiện đúng một lần. Cần xóa ít nhất bao nhiêu phần tử trong *v* để *v* trở thành vector beautifull.
    - + Với *v* = [2, 3, 6, 3] thì vectorBeautifull(v) = 1
      → chỉ cần xóa số 3.
    - + Với *v* = [1, 2, 3] thì vectorBeautifull(v) = 0

Sử dụng set *s*, chèn tất cả phần tử trong *v* vào *s*.

Kết quả chính là v.size() - s.size().

Cấu trúc dữ liệu tập hợp (set)

- Ứng dụng

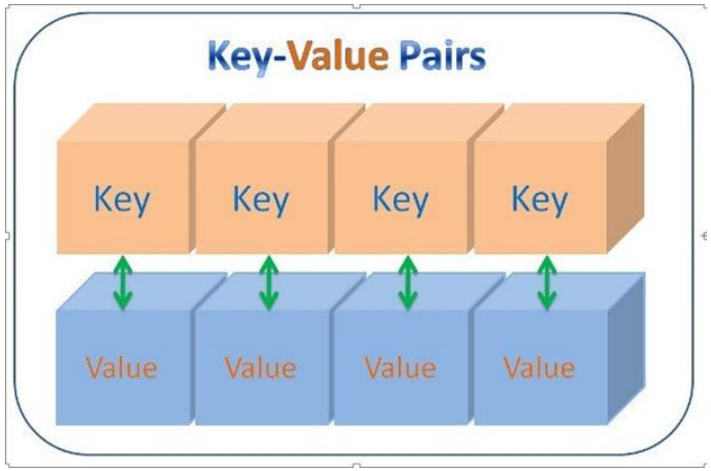
  Ví dụ 3: Viết chương trình liệt kê các số nguyên tố trong phạm vi từ 1 đến n theo phương pháp Eratosthene (không cần đến các phép nhân).

  Thuật toán: xuất phát từ một tập các số nguyên từ 1 đến n. Mỗi lần gặp một số nguyên tố, ta sẽ loại trừ ra khỏi tập này tất cả các số là bội của số nguyên tố này.

  (Tương truyền rằng Eratosthene là một nhà bác học thời chưa có giấy viết. Vì vậy bài toán tìm số nguyên tố được ông sáng tác và thực hiện trên mặt đất: ông kẻ một dãy ô, trong đó ghi các số tự nhiên từ 1 đến n. Sau đó, số nào không phải là số nguyên tố thì bị loại ra bằng cách lấy que chọc một lỗ vào đó để xóa nó đi. Kết quả trên mặt đất lỗ chỗ như một cái sàng, sàng lọc các số)

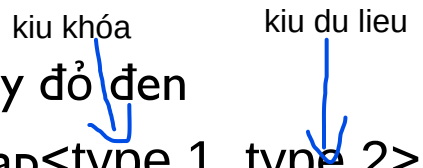
Cấu trúc dữ liệu ánh xạ (map)

- Khái niệm
  - Dữ liệu có kiểu cấu trúc liên kết
  - Là sự kết hợp của khóa (key value) và ánh xạ của nó (mapped value)
  - Giá trị các khóa
    - Duy nhất (không trùng)
    - Map
      - Được sắp thứ tự tăng dần
    - Unordered_map
      - Không được sắp
  - *Chú ý*: trong Python, C# thì map là dictionary



Cấu trúc dữ liệu ánh xạ (map)

- Cài đặt
  - Trong C++
    - map
      - Dựa trên cây đỏ đen
      - Khai báo: map<type 1, type 2> tien_bien;
      - Lấy giá trị khóa (key): tien_bien.first
      - Lấy giá trị (value): ten_bien.second
    - unordered_map
      - Dựa trên bảng băm (hash table)
      - Khai báo: unordered_map<type 1, type 2> map_name;



Cấu trúc dữ liệu ánh xạ (map)

- Cài đặt
  - Thay đổi thứ tự sắp xếp các phần tử của map

```
struct cmp {
    bool operator() (char a, char b) {
        return a > b;
    }
};
.....
map <char, int, cmp> m;
```

Cấu trúc dữ liệu ánh xạ (map)

- Cài đặt
  - Các hàm cơ bản
    - value = at(k): trả lại giá trị ứng với khóa k
    - size(): số phần tử hiện tại
    - max_size(): số phần tử mà map có thể chứa
    - empty(): kiểm tra map có rỗng không
    - erase(const g): xóa khóa g khỏi map
    - pair insert(keyvalue, mapvalue)
    - clear(): xóa tất cả các phần tử

Cấu trúc dữ liệu ánh xạ (map)

- Cài đặt
  - Ví dụ
    - Chèn dữ liệu
      - map_name. insert(pair<int, int>(160, 42));
    - Xóa
      - map_name.erase(160);
    - Duyệt

```
map<int, int>::iterator itr;
for (itr = map_name.begin(); itr != map_name.end(); ++itr) {
        cout  << itr.first << "t   t" << itr.second << endl;
}
```

Cấu trúc dữ liệu ánh xạ (map)

- Ứng dụng
  - Tìm chủ thuê bao theo số điện thoại
  - Tìm địa chỉ IP theo tên miền
  - Tìm số lần vi phạm kỷ luật của một nhân viên

Cấu trúc dữ liệu ánh xạ (map)

- Bài tập:
  - Bài 1: Cho một chuỗi *s*, hãy đưa ra một dãy lần lượt là các ký tự và số lần xuất hiện của nó và các ký tự được sắp xếp theo thự tự từ điển.

Cấu trúc dữ liệu ánh xạ (map)

- Bài tập:
  - Bài 2: Cho danh sách các sản phẩm của 2 kho hàng A và B. Do chiến lược kinh doanh bạn được giao nhiệm vụ nhập các sản phẩm từ kho B vào kho A sao cho những sản phẩm nào đã có trong kho A thì không nhập.
  - Ví dụ: Với A = {"Banana", "Banana", "Apple“},
    B = {"Orange", "Apple", "Banana", "Watermelon“},
    thì mergeProducts(A, B) = {Apple → false, Banana → false, Orange → true, Watermelon → true}

TỔNG KẾT