

Tập hợp

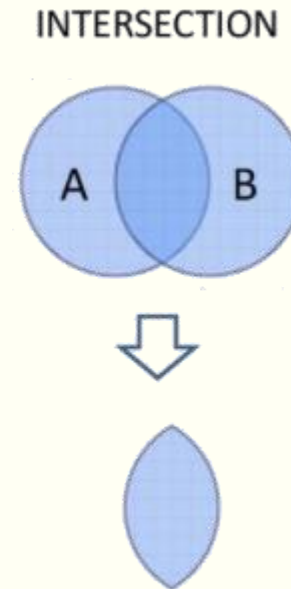
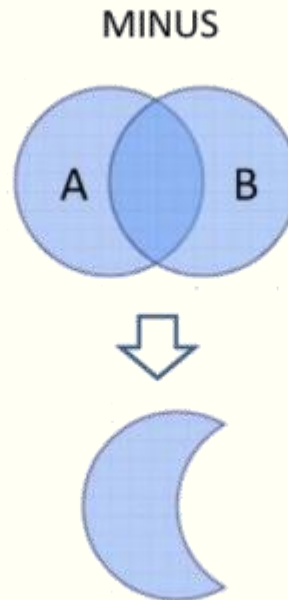
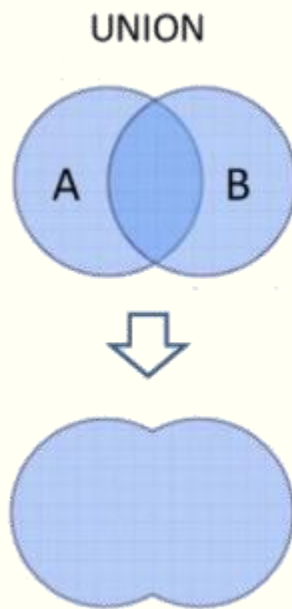


Khái niệm

- Tập hợp lưu một danh sách các phần tử cùng kiểu dữ liệu (thường không quan tâm đến thứ tự)
- Các phần tử khác nhau



Các phép toán



Phép hợp

➤ Xét hai tập hợp A và B. Hợp của A và B là tập hợp gồm các phần tử thuộc A, B và các phần tử giao giữa chúng.

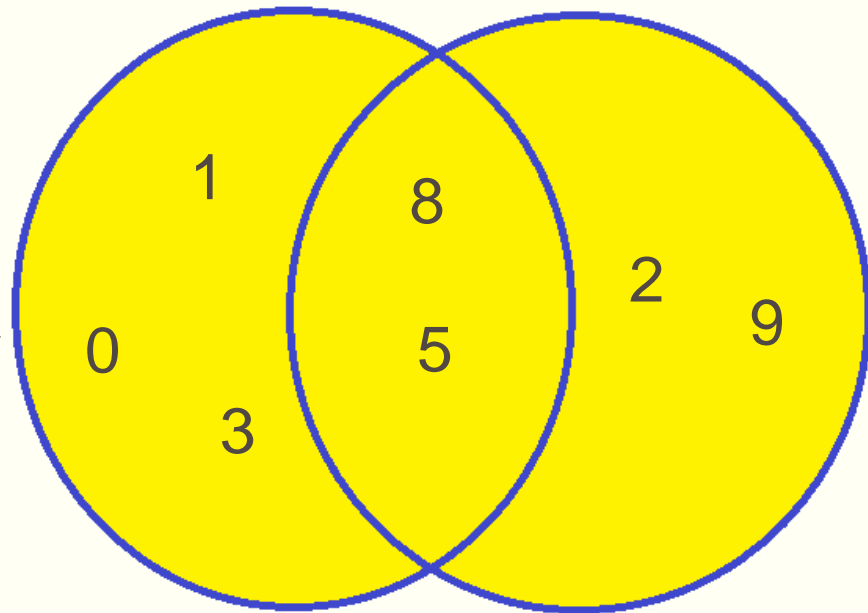
➤ Ví dụ:

$$A = \{0, 1, 3, 5, 8\}$$

$$B = \{2, 5, 8, 9\}$$

$$A \cup B = \{0, 1, 2, 3, 5, 8, 9\}$$

$$A \cup B = B \cup A$$



Phép giao

➤ **Definition:** Xét hai tập hợp A và B. Giao của A và B là tập hợp gồm các phần tử thuộc A và B.

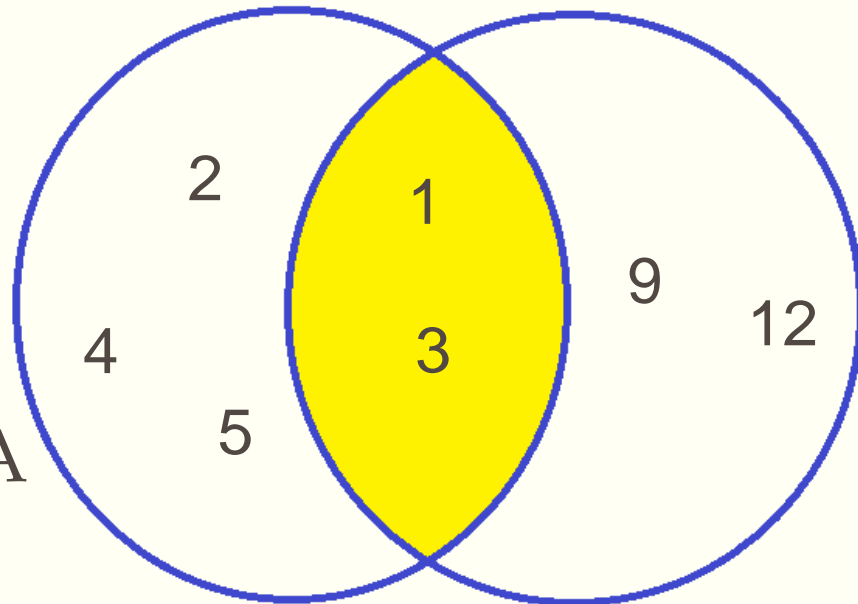
➤ Ví dụ:

$$A = \{1, 2, 3, 4, 5\}$$

$$B = \{1, 3, 9, 12\}$$

$$A \cap B = \{1, 3\}$$

$$\text{Note: } A \cap B = B \cap A$$



Phép hiệu

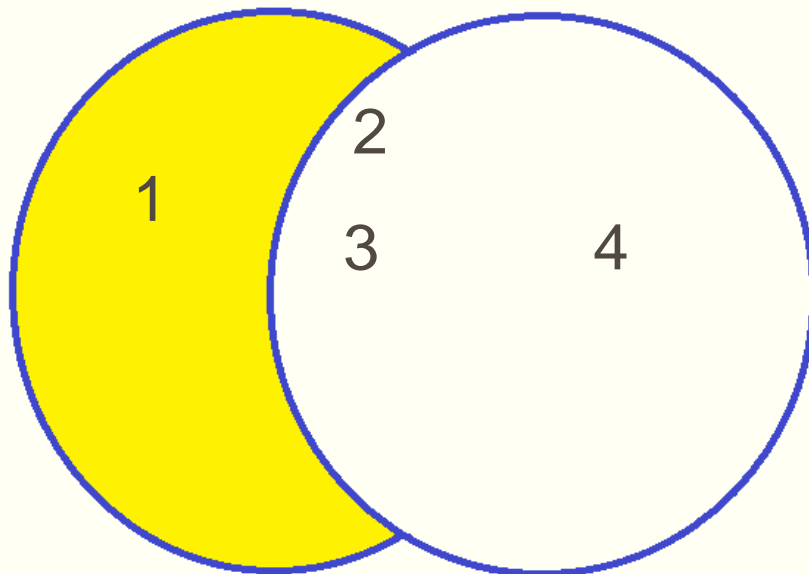
➤ **Definition:** Xét hai tập hợp A và B. Hiệu của A và B là tập hợp gồm các phần tử thuộc A mà không thuộc B.

➤ Ví dụ:

$$A = \{1, 2, 3\}$$

$$B = \{2, 3, 4\}$$

$$A - B = \{1\}$$



Thư viện Set

```
// unordered_set::insert
#include <iostream>
#include <string>
#include <array>
#include <unordered_set>
int main () {
    std::unordered_set<std::string> myset = {"yellow","green","blue"};
    std::array<std::string,2> myarray = {"black","white"};
    std::string mystring = "red";

    myset.insert (mystring); // copy insertion
    myset.insert (myarray.begin(), myarray.end()); // range insertion
    myset.insert ( {"purple","orange"} ); // initializer list insertion

    std::cout << "myset contains:";
    for (const std::string& x: myset) std::cout << " " << x;
    std::cout << std::endl;

    return 0;
}
```

Bài tập

1. Cho dãy n phần tử. Hãy đếm số phần tử khác nhau trong mảng
2. Cho danh sách chứa thời gian quét thẻ của nhân viên trong ngày. Hãy kiểm tra xem có bao nhiêu người khác nhau đi làm trước 9h sáng

- Cấu trúc tập hợp được sử dụng trong một số bài toán liên quan đến việc lưu trữ các phần tử dữ liệu duy nhất.