

# Tin học cơ sở 4

## Buổi 5. Mạng

Bộ môn Khoa học máy tính - 2017



# Nội dung buổi học

---

1. Khai báo mạng, khởi tạo mạng
2. Truy xuất mạng
3. Duyệt mạng

# Dữ liệu

- Một biến lưu một dữ liệu
  - ❖ Phải khai báo nhiều lần cho nhiều dữ liệu

```
int x = 4;
```

4

```
int y = 7;
```

7

```
int z = 0;
```

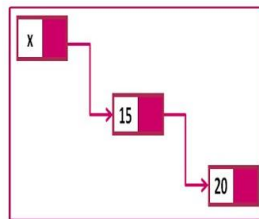
0

# Dữ liệu lớn

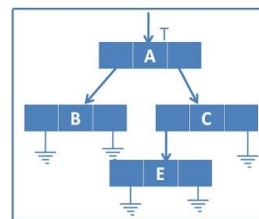
- Lưu trữ và xử lý khối lượng dữ liệu lớn
  - ❖ Cần các cấu trúc dữ liệu phức tạp hơn



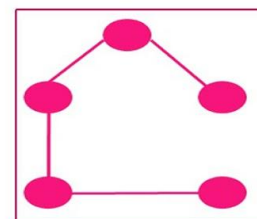
Sorting



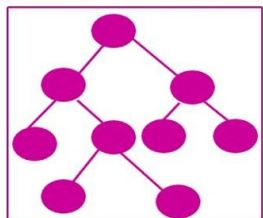
Link list



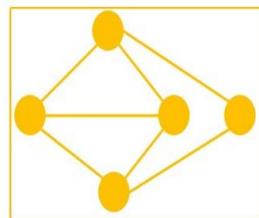
list



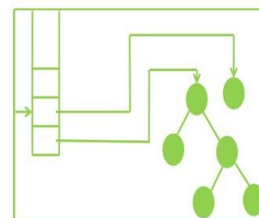
spanning tree



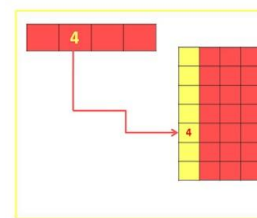
Tree



Graph



Stack



Hashing

By...navinkumardhoprephotography.com

# Mảng

- Mảng: một dãy các **giá trị cùng kiểu (phần tử)** được đánh **chỉ mục**

|   |       |       |     |
|---|-------|-------|-----|
| 4 | Hello | true  | 'a' |
| 7 | World | false | 'b' |
| 0 | !!!   | false | 'c' |
|   |       | true  |     |

# Mảng

---

➤ Ví dụ:

- ❖ 52 quân bài
- ❖ 4000 sinh viên của Trường Công nghệ
- ❖ 1.000.000 tài khoản Facebook
- ❖ 4 tỷ nucleotid trong chuỗi ADN
- ❖ 73 tỷ yêu cầu tìm kiếm Google mỗi năm

# Khai báo mảng

➤ Mảng tĩnh (số phần tử cố định)

<tên kiểu phần tử> <tên mảng>[<số phần tử>];

➤ Ví dụ:

❖ `int numbers[10];` // mảng 10 số nguyên

❖ `string names[100];` // mảng 100 chuỗi ký tự

❖ `double angles[1000];` // mảng 1000 số thực

# Khai báo mảng

- Mảng động dùng thư viện `<vector>`  
`vector<ten kiểu phần tử> <ten mảng>(<số phần tử>, <giá trị khởi tạo>);`
- ❖ Thư viện `#include <vector>`
- Ví dụ
  - ❖ `vector<int> numbers(100, 0);` // mảng 100 số nguyên khởi tạo bằng số 0
  - ❖ `vector<string> names(100);` // mảng 100 chuỗi (xâu rỗng)

# Truy xuất mảng

➤ Các phần tử được đánh số từ 0

❖ `int a[100]`: đánh số từ 0 đến 99

• Giống số nhà

❖ Phần tử thứ nhất là `a[0]`

❖ Phần tử thứ hai là `a[1]`

❖ ...

❖ Phần tử cuối cùng là `a[99]`

➤ Các phần tử xếp liên tục trong bộ nhớ

`a[99]`

`a[98]`

...

`a[2]`

`a[1]`

`a[0]`

# CT1. Khởi tạo mảng / in giá trị

```
int main()
{
    const int N = 10;
    int num[N]; // mảng num có 10 số nguyên

    for (int i = 0; i < N; i++) {
        num[i] = 0; // khởi tạo bằng số 0
    }

    for (int i = 0; i < N; i++) {
        cout << num[i] << endl; // lần lượt in các số
    }
    return 0;
}
```

## CT2. Khởi tạo mảng liên tiếp

```
int start = 2;  
for (int i = 0; i < N; i++) {  
    num[i] = start+i;  
}
```

# CT2'. Khởi tạo mảng ngẫu nhiên

```
//mảng real có 1000 số thực
const int N_double = 1000;
double real[N_double];

for (int i = 0; i < N_double; i++) {
    real[i] = 0.5; // khởi tạo với số 0.5
}

for (int i = 0; i < N_double; i++) {
    real[i] = 1.0 * rand() / RAND_MAX;
}
```

## CT3. Tính tổng các số

```
int sum = 0;
for (int i = 0; i < N; i++) {
    sum += num[i];
}
cout << sum << endl;
```

# CT4. Các số Fibonacci

```
num[0] = 1;
```

```
num[1] = 1;
```

```
for (int i = 2; i < N; i++) {  
    num[i] = num[i-1] + num[i-2];  
}
```

## CT5. Nhập mảng

```
// mảng str có 100 xâu kí tự  
const int N_str = 100;  
string str[N_str];  
  
for (int i = 0; i < N_str; i++) {  
    cin >> str[i];  
}
```

## CT6. Sao chép mảng

```
string strCopy[N_str];  
  
for (int i = 0; i < N_str; i++) {  
    strCopy[i] = str[i];  
}
```

# CT7. Tích vô hướng 2 véc-tơ

```
double v1[N], v2[N];  
// nhập 2 véc-tơ hoặc khởi tạo chúng  
...  
  
// tích vô hướng  
double dot = 0;  
for (int i = 0; i < N; i++) {  
    dot += v1[i] * v2[i];  
}  
cout << dot << endl;
```

## CT8. Tổng 2 véc-tơ

```
double va[N], vb[N], vc[N];  
// khởi tạo hoặc nhập va và vb  
...  
// tính tổng 2 véc-tơ va và vb  
// kết quả đặt ở vc  
for (int i = 0; i < N; i++) {  
    vc[i] = va[i] + vb[i];  
}
```

# Duyệt mảng

---

- Duyệt từng phần tử
  - ❖ Tính tổng tích lũy
  - ❖ Tìm kiếm
- Duyệt từng cặp phần tử
  - ❖ Vòng lặp lồng nhau
- Duyệt các cặp phần tử thỏa mãn điều kiện

# CT9. Tìm kiếm theo điều kiện

```
for (int i = 0; i < N_double; i++) {  
    // tìm các số nhỏ hơn 0.5  
    if (real[i] < 0.5) {  
        cout << i << " " << real[i] << endl;  
    }  
}
```

# CT10. Tìm kiếm cặp số

```
for (int i = 0; i < N; i++) {  
    for (int j = i+1; j < N; j++) {  
        // tìm các cặp số có một số  
        // là bội số của số kia  
        if ( (num[j] != 0 && num[i] % num[j] == 0) ||  
            (num[i] != 0 && num[j] % num[i] == 0) ) {  
            cout << "(" << num[i] << ", "  
                << num[j] << ")" << endl;  
        }  
    }  
}
```

# Các lỗi thường gặp

- Sử dụng chỉ số không hợp lệ
  - ❖ Chỉ số hợp lệ:  $\{0, 1, \dots, N-1\}$
  - ❖ Phải kiểm tra chỉ số hợp lệ trước khi sử dụng chỉ số để truy xuất mảng
- Không khởi tạo/nhập mảng trước khi truy xuất

# Câu hỏi

---

- Về nội dung buổi học ?
- Về nội dung khóa học ?
- Về cách tổ chức ?