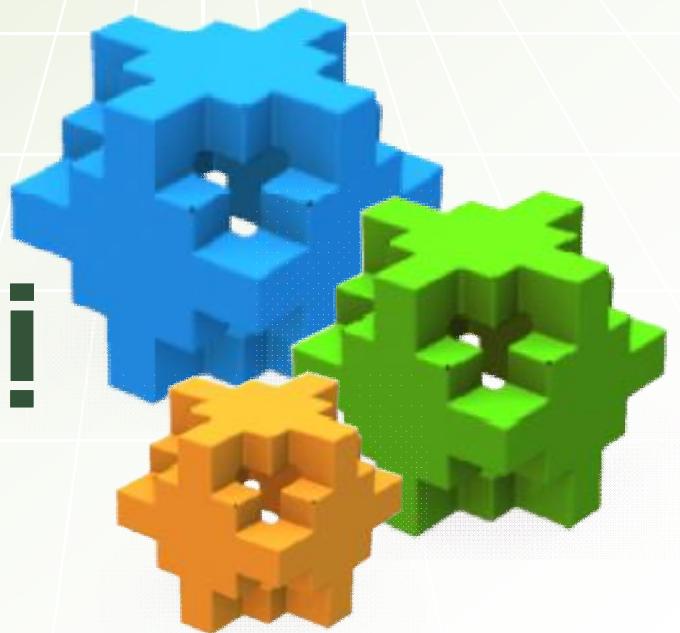


LOGO

Lý thuyết đồ thị

Phạm Thị Vương





Nội dung

1

Thông tin môn học

2

Tài liệu tham khảo

3

Đánh giá kết quả

4

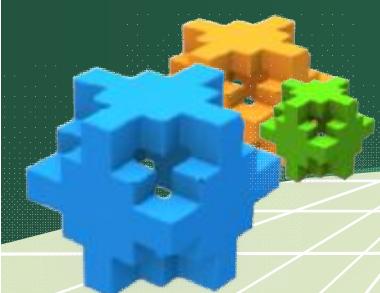
Nội dung môn học

5

Nội dung thực hành

6

Ôn tập



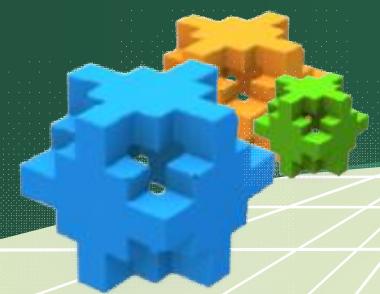
Thông tin môn học

- ✓ Tên: Lý thuyết đồ thị
- ✓ Số tín chỉ: 4
- ✓ Lý thuyết: 45; Thực hành 30
- ✓ Website: <http://ltdt.ptvuong.com/>
- ✓ Mail: lythuyetdothi@ptvuong.com



Tài liệu tham khảo

- ✓ Lý thuyết đồ thị, Nguyễn Cam & Chu Đức
Khánh, 1998, NXB Trẻ
- ✓ Toán rời rạc nâng cao, Trần Ngọc Danh,
2004, NXB ĐHQG TPHCM
- ✓ Algorithms, Robert Sedgewich, 1983,
Addison Wesley
- ✓ Toán rời rạc ứng dụng trong Tin Học –
Kenneth H.Rosen, NXB KHKT - 1997



Đánh giá kết quả

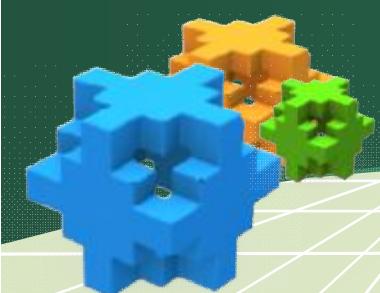
- ✓ Bài thi cuối kỳ: 7đ – không dùng tài liệu
- ✓ Thực hành: 3 đ
- ✓ Ngôn ngữ thực hành: C/C++
- ✓ Công cụ: Dev C++
<http://www.bloodshed.net/devcpp.html>



Nội dung môn học

✓ Chương 1: Đại cương về đồ thị

- § Định nghĩa đồ thị
- § Một số khái niệm trên đồ thị
- § Đường đi, chu trình, tính liên thông
- § Duyệt đồ thị
- § Biểu diễn đồ thị trên máy tính



Nội dung môn học

✓ Chương 2: Các bài toán về đường đi ngắn nhất

§ Định nghĩa, phát biểu bài toán

§ Thuật toán Dijkstra

§ Thuật toán Ford – Bellman

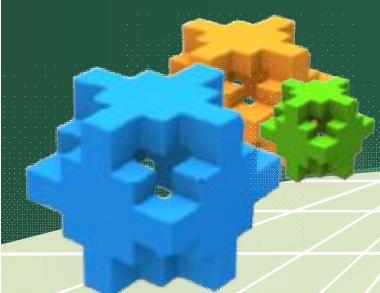
§ Thuật toán Floy



Nội dung môn học

✓ Chương 3: Cây

- § Các khái niệm cơ bản
- § Cây bao trùm(Spanning tree)
- § Thuật giải Prim
- § Thuật giải Krusal
- § Cây nhị phân
- § Cây nhị phân tìm kiếm

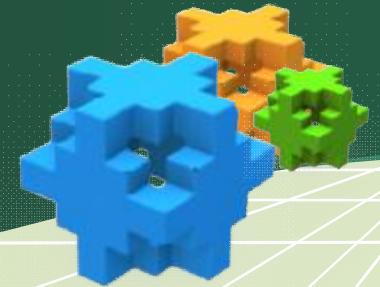


Nội dung môn học

✓ Chương 4: Chu trình Euler và Hamilton

§ Chu trình Euler, Thuật toán Fleury để
tìm chu trình Euler

§ Chu trình Hamilton, Quy tắc tìm chu
trình Hamilton



Nội dung thực hành

- ✓ DFS, BFS, kiểm tra liên thông, tìm thành phần liên thông
- ✓ Dijkstra
- ✓ Floy
- ✓ Belman
- ✓ Prim
- ✓ Krusal
- ✓ Euler



Lưu ý về phong cách lập trình

- ✓ Đặt tên
- ✓ Tab
- ✓ Khai báo prototype trước main()
- ✓ {}



Ôn tập

- ✓ Kỹ năng lập trình C: đọc ghi file, đệ quy
- ✓ Nắm các cấu trúc dữ liệu: mảng 2 chiều, stack, queue, cây



1 bài tập C

✓ Nhập 4 biến nguyên và xuất các giá trị
vừa nhập



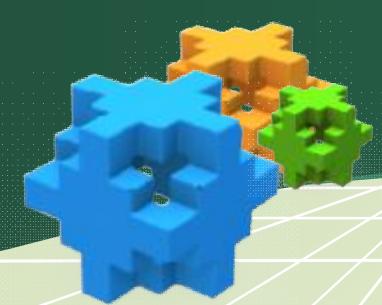
Giải

1. Dùng 4 biến => cách dài nhất, cơ bản nhất
2. Dùng mảng => khai báo biến gọn hơn, 1 lần thay cho nhiều lần
3. Dùng mảng và vòng lặp do while => viết code nhập gọn hơn, viết 1 lần thay cho nhiều lần
4. Dùng mảng và vòng lặp for => viết code gọn hơn, vì vòng for viết gọn hơn vòng while
5. Dùng mảng, vòng lặp for gộp => viết code gọn hơn, nhưng không tách riêng được 2 phần nhập xuất



Giải

6. Dùng hàm (function) để tách riêng phần nhập xuất => code có thể tái sử dụng nhiều lần
7. Dùng file để nhập xuất từ file thay cho việc nhập bằng bàn phím và xuất ra màn hình



Cách 1:

```
int a1, a2, a3, a4;  
printf("\n a1 = ");  
scanf("%d", &a1);  
printf("\n a2 = ");  
scanf("%d", &a2);  
printf("\n a3 = ");  
scanf("%d", &a3);  
printf("\n a4 = ");  
scanf("%d", &a4);  
printf("\n Ban vua nhap vao 4 so: %d %d %d %d", a1, a2, a3, a4);
```

```
a1 = 3  
a2 = 4  
a3 = 7  
a4 = 9  
Ban vua nhap vao 4 so: 3 4 7 9_
```



Cách 2:

```
int a[4];
printf("\n a1 = ");
scanf("%d", &a[0]);
printf("\n a2 = ");
scanf("%d", &a[1]);
printf("\n a3 = ");
scanf("%d", &a[2]);
printf("\n a4 = ");
scanf("%d", &a[3]);

printf("\n Ban vua nhap vao 4 so: %d %d %d %d", a[0], a[1], a[2], a[3]);
```

Cách 3: dùng vòng lặp do while

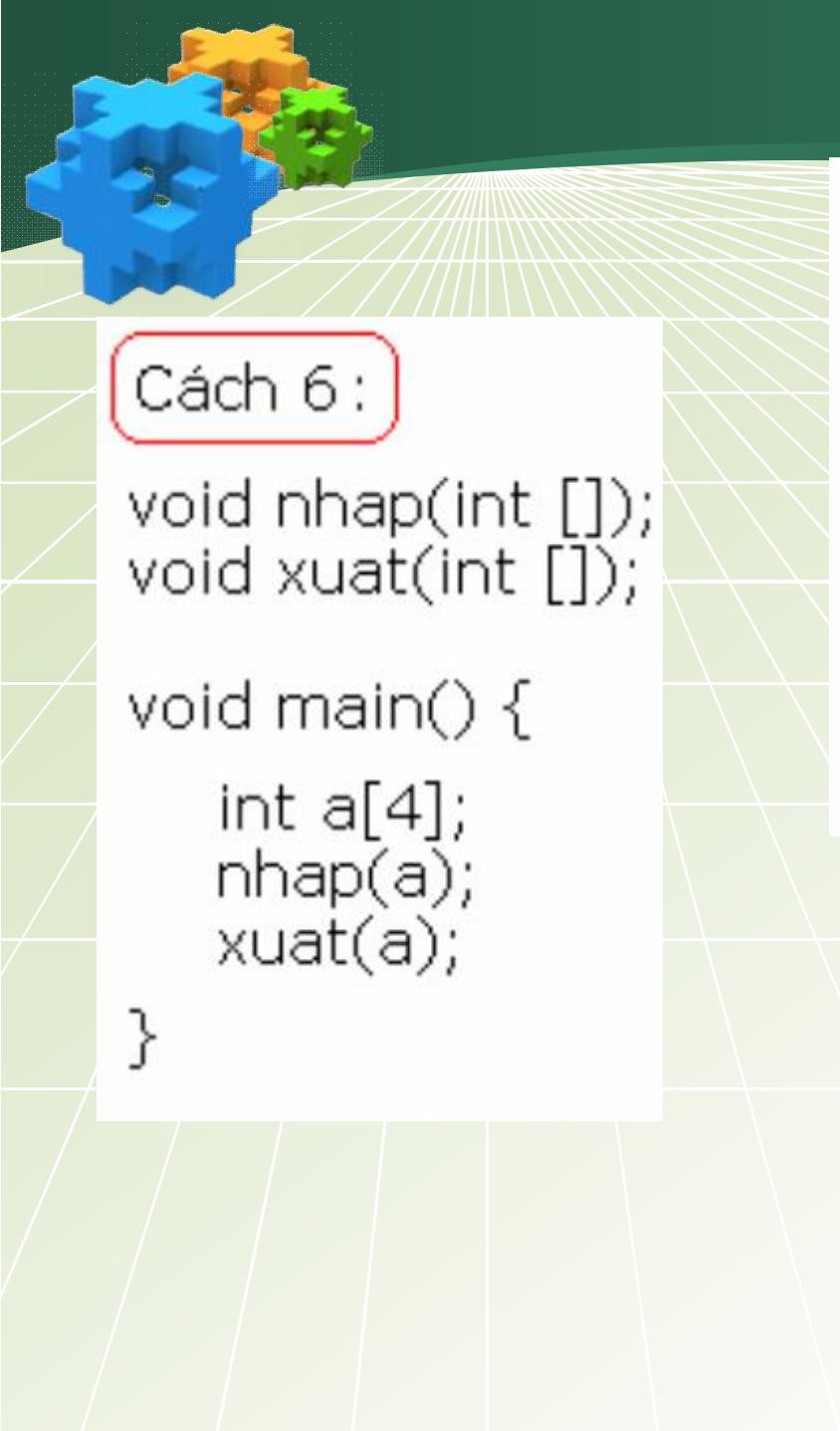
```
int a[4], i;  
i=0;  
do  
{  
    printf("\n a%d = ", i);  
    scanf("%d", &a[i]);  
    i++;  
} while (i<4);  
  
i=0;  
do  
{  
    printf("%d ", a[i]);  
    i++;  
} while (i<4);
```

Cách 4: dùng vòng lặp for

```
int a[4], i;  
for (i=0; i<4; i++) {  
    printf("\n a%d = ", i);  
    scanf("%d", &a[i]);  
}  
  
printf("\n Ban vua nhap vao 4 so: ");  
for (i=0; i<4; i++) {  
    printf("%d ", a[i]);  
}
```

Cách 5:

```
int a[4], i;  
for (i=0; i<4; i++) {  
    printf("\n a%d = ", i);  
    scanf("%d", &a[i]);  
    printf("%d ", a[i]);  
}
```

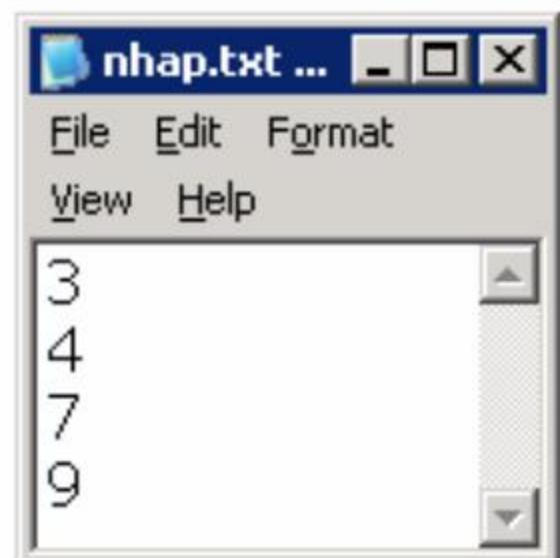


Cách 6 :

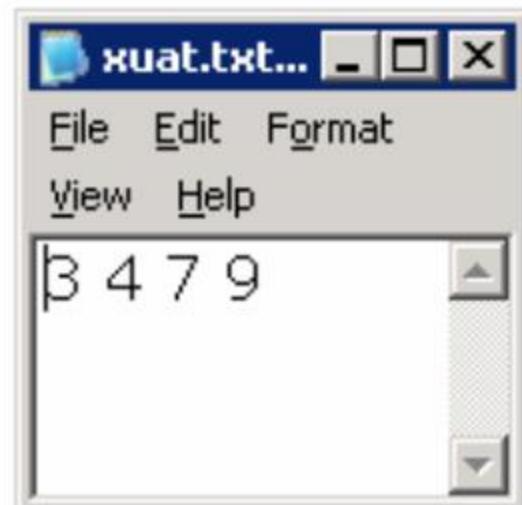
```
void nhap(int []);  
void xuat(int []);  
  
void main() {  
    int a[4];  
    nhap(a);  
    xuat(a);  
}
```

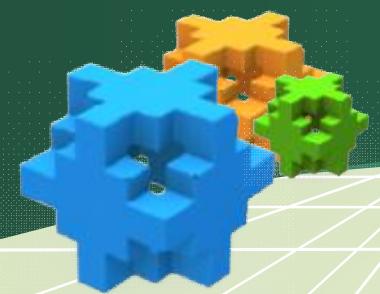
```
void nhap(int b[]) {  
    int i;  
    for (i=0; i<4; i++) {  
        printf("\n a%d = ", i);  
        scanf("%d", &b[i]);  
        printf("%d ", b[i]);  
    }  
}  
  
void xuat(int c[]) {  
    printf("\n 4 so: ");  
    for (i=0; i<4; i++) {  
        printf("%d ", c[i]);  
    }  
}
```

```
void nhap(int b[], char *f) {
    int i;
    FILE *fp;
    fp = fopen(f, "r");
    for (i=0; i<4; i++) {
        fscanf(fp, "%d", &b[i]);
    }
    fclose(fp);
}
```



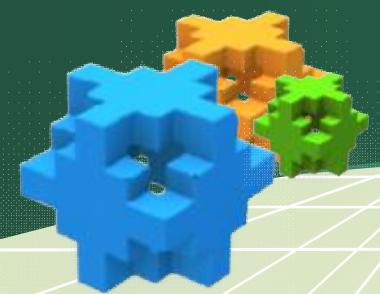
```
void xuat(int c[], char *f) {
    int i;
    FILE *fp;
    fp = fopen(f, "w");
    for (i=0; i<4; i++) {
        fprintf(fp, "%d ", c[i]);
    }
    fclose(fp);
}
```





Ôn tập

- ✓ FILE *vb, *np
- ✓ FILE *fopen(const char *tên_tệp, const char *kiểu);
- ✓ int fclose(FILE *fp);
- ✓ int feof(FILE *fp);
- ✓ fscanf(FILE *fp, chuoidinhdang, cac bien);
- ✓ fprintf(FILE *fp, chuoidinhdang, cac bien);



Bài tập ôn

- ✓ Tìm số lớn nhất, nhỏ nhất trong mảng.
- ✓ Đếm số phần tử thỏa điều kiện nào đó
- ✓ Sắp xếp mảng.
- ✓ Ngăn xếp.
- ✓ Hàng đợi.

LOGO

Thank You !

