

# ĐỀ THI: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

(Thời gian 90')

Mã đề CDK54-2010-02

Sinh viên được sử dụng tài liệu

## Bài 1.

- a) Trình bày sự khác biệt giữa mảng cấp phát bộ nhớ động và mảng cấp phát tĩnh? Khi nào dùng mảng cấp phát động, mảng cấp phát tĩnh, cho ví dụ.

Xem trong vở

- b) Đánh giá thời gian thực hiện của thuật toán tính mũ nhanh sau theo mô hình O-lớn

```
double fastPower(double x, int n)
{
    double fract;
    if(n==0) return 1;
    fract = fastPower(x,n/2);

    if(n%2==0) return fract*fract;
    else return fract*fract*x;
}
```

Thuật toán trên sẽ thực hiện việc tính  $x^n$

Thuật toán này là thuật toán đệ quy, ta có công thức đệ quy là

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

Thời gian thực hiện theo O-lớn là  $O(\log n)$

Chú ý: Có thể dùng 1 trong 3 phương pháp đã học để chứng minh!

## Bài 2.

- a) Trong các phương pháp sắp xếp: lựa chọn, chèn, đổi chỗ (nổi bọt), quicksort (sắp xếp nhanh), mergesort (sắp xếp trộn), thì phương pháp nào là phù hợp nhất để sắp xếp trên mảng với số lượng các phần tử lớn? Giải thích lựa chọn của bạn.

QuickSort là phù hợp nhất để sắp xếp trên mảng với số lượng phần tử lớn vì:

- Thời gian cỡ  $O(n \log n)$
- So với mergesort thì nó không cần sử dụng thêm bộ nhớ phụ
- Không cần thực hiện thao tác trộn

- b) Cho mảng  $A=\{14,45,21,12,7,87,25,56,91,64,33,41,72,89,62\}$

Hãy minh họa lần lặp thứ nhất của thuật toán sắp xếp nổi bọt trên A.

Tùy từng trường hợp sắp xếp tăng dần hoặc giảm dần mà kết quả sẽ khác nhau. Trong trường hợp tăng dần thì lần lặp đầu tiên, khóa có giá trị lớn nhất sẽ bị đẩy về cuối dãy.

Cụ thể xem ví dụ trong slide

**Bài 3.**

- a) Hàng đợi vòng (Circle Queue) là gì ? Tác dụng của việc tổ chức hàng đợi dạng vòng?

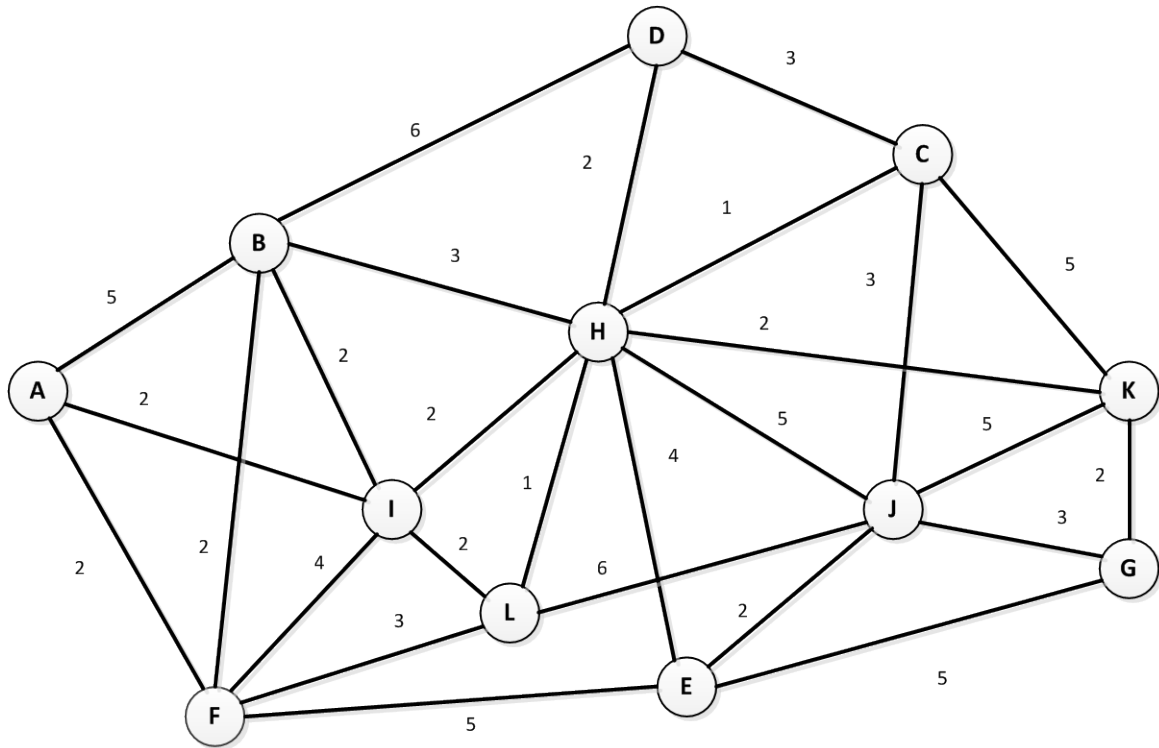
Hàng đợi vòng và tác dụng của hàng đợi vòng : Xem trong slide vì đây là phần lý thuyết căn bản

- b) Áp dụng thuật toán định giá biểu thức hậu tố bằng stack để tính giá trị biểu thức dạng hậu tố sau (cần nêu rõ các bước trung gian trong quá trình tính)

$$15\ 2\ 3\ +\ /\ 2\ ^\ 4\ 12\ 2\ -\ \% \ +$$

Kết quả là 13, còn trình tự thực hiện xem các ví dụ đã chữa trên lớp

**Bài 4.** Cho đồ thị



- Biểu diễn đồ thị dùng ma trận kề
- Thực hiện thuật toán PRIM và đưa ra cây khung có trọng số nhỏ nhất trên đồ thị (cần mô tả rõ các bước thực hiện)

Cây khung trọng số nhỏ nhất là 21  
Trình tự thực hiện xem các ví dụ đã chữa trên lớp

**Bài 5.** Viết hàm nhận đầu vào là 1 ma trận kề biểu diễn cho 1 đồ thị vô hướng, và số lượng đỉnh trên đồ thị. Hàm kiểm tra xem đồ thị có liên thông hay không. Nếu đồ thị liên thông thì hàm trả về giá trị 1, ngược lại thì hàm trả về giá trị 0.

```
int ktLienThong(int Adj[100][100], int n)
{
    //Thân hàm
}
```

Trong đó  $n$  là số lượng đỉnh thực sự trên đồ thị ( $0 < n \leq 100$ ), Adj là ma trận kề lưu trữ đồ thị.

Chú ý : đồ thị liên thông nếu mà giữa hai cặp đỉnh bất kỳ luôn tồn tại đường đi.

Hãy đưa ra đánh giá thời gian thực hiện của thuật toán của bạn theo O-lớn

**Chú ý: chương trình sử dụng thuật toán tối ưu hơn sẽ được đánh giá cao hơn!**