

Đề thi giữa kì*Thời gian làm bài: 90 phút*

Câu 1. Vì sao để đánh giá thuật toán nên dùng phân tích toán học hơn phân tích thực nghiệm? Hãy đưa ra 2 đoạn chương trình ví dụ có thời gian $O(1)$.

Câu 2. Giải thích mối quan hệ giữa $f(n)$ và $g(n)$ trong biểu thức $f(n) = O(g(n))$. Hãy đưa ra 2 đoạn chương trình ví dụ có thời gian $O(n)$.

Câu 3. Khi áp dụng thuật toán tìm kiếm nhị phân để tìm $x = 3$ trên mảng a (hình dưới), ta cần bao nhiêu phép so sánh x với phần tử của a ?

a

2	5	7	11	12	20	22	30	104
---	---	---	----	----	----	----	----	-----

Câu 4. Khi áp dụng thuật toán tuyến tính trộn 2 mảng a, b tăng thành một mảng c tăng (hình dưới), ta cần bao nhiêu phép so sánh phần tử của a với phần tử của b ?

a

2	22	30
---	----	----

b

5	7	11	12	20	104
---	---	----	----	----	-----

c

2	5	7	11	12	20	22	30	104
---	---	---	----	----	----	----	----	-----

Câu 5. Phân tích độ phức tạp thời gian của đoạn mã sau:

```
int mystery1(int n){
    int sum = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            sum += i * j;
        }
    }
    return sum;
}
```

Câu 6. Phân tích độ phức tạp thời gian của đoạn mã sau:

```
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++){
        c[i][j] = 0;
        for (int k = 0; k < n; k++)
            c[i][j] += a[i][k] * b[k][j];
    }
```

Câu 7. Phân tích độ phức tạp thời gian của đoạn mã sau:

```
int mystery2(int a[], unsigned int n)
{
    int sum = 0;
    int thisSum = 0;
    for(int i = 0; i < n; i++){
        thisSum += a[i];
        if(thisSum > sum)
            sum = thisSum;
        if(thisSum < 0)
```

```

        thisSum = 0;
    }
    return sum;
}

```

Câu 8. Cho biết a là một mảng n phần tử và swap là hàm trao đổi giá trị 2 đối số. Đoạn mã dưới đây sắp xếp mảng a theo thứ tự giảm dần:

```

for(int i = 0; i < n - 1; i++)
    for(int j = 0; j < n - 1 - i; j++)
        if(a[j] < a[j + 1]) swap(a[j], a[j + 1]);

```

Hãy cho biết đoạn mã trên gọi hàm swap bao nhiêu lần nếu truyền vào mảng a có $a[i] = i$ với $i = 0, \dots, n-1$?

Câu 9. Mô tả ngắn gọn và so sánh các phương án cài đặt ngăn xếp nêu dưới đây:

- (a) cài bằng mảng tĩnh
- (b) cài bằng danh sách liên kết đơn
- (c) cài bằng danh sách liên kết kép

Câu 10. Bắt đầu từ ngăn xếp rỗng, ta thực hiện xen lẫn các phép push và các phép pop trên nó. Các phép push lần lượt thêm các số nguyên từ 0 đến 9 vào ngăn xếp. Mỗi phép pop loại một phần tử khỏi ngăn xếp và in giá trị vừa loại ra màn hình. Chuỗi output nào sau đây không thể xảy ra trên màn hình? Giải thích.

- (a) 4 3 2 1 0 9 8 7 6 5
- (b) 4 6 8 7 5 3 2 9 0 1
- (c) 2 5 6 7 4 8 9 3 1 0
- (d) 4 3 2 1 0 5 6 7 8 9
- (e) 1 2 3 4 5 6 9 8 7 0
- (f) 0 4 6 5 3 8 1 7 2 9
- (g) 1 4 7 9 8 6 5 3 0 2
- (h) 2 1 4 3 6 5 8 7 9 0

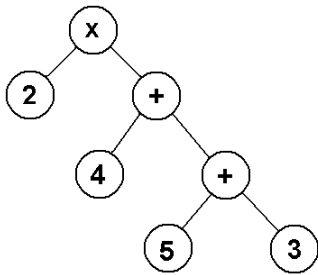
Câu 11. Giả sử ta thực hiện trộn lẫn enqueue và dequeue trên một hàng đợi. Ban đầu hàng đợi rỗng. Các phép enqueue lần lượt thêm các số nguyên từ 0 đến 9 vào hàng đợi này. Phép dequeue loại một phần tử khỏi hàng đợi và in giá trị vừa loại ra màn hình. Chuỗi output nào sau đây không thể xảy ra trên màn hình? Giải thích.

- (a) 0 1 2 3 4 5 6 7 8 9
- (b) 4 6 8 7 5 3 2 9 0 1
- (c) 2 5 6 7 4 8 9 3 1 0
- (d) 4 3 2 1 0 5 6 7 8 9

Câu 12. Mô tả ngắn gọn và so sánh các phương án cài đặt hàng đợi nêu dưới đây:

- (a) cài bằng mảng tĩnh vòng
- (b) cài bằng danh sách liên kết đơn
- (c) cài bằng danh sách liên kết kép.

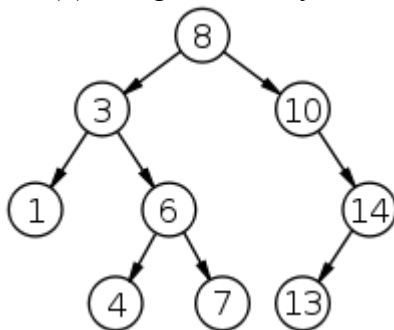
Câu 13. Viết biểu thức postfix cho cây biểu thức số học bên dưới. Trình bày và phân tích thuật toán tính giá trị biểu thức postfix.



Câu 14. Hãy vẽ cây nhị phân biểu diễn biểu thức số học postfix 1 5 12 7 4 - / * +. Trình bày và phân tích thuật toán xây dựng cây biểu thức từ dạng postfix.

Câu 15. Cho cây tìm kiếm nhị phân trong hình vẽ. Hãy cho biết cây kết quả sau khi

- (a) thêm 11 vào cây
- (b) xóa gốc khỏi cây thu được ở câu a



Câu 16.

- (a) Bắt đầu từ cây tìm kiếm nhị phân rỗng, lần lượt thêm vào dãy khóa 16 6 2 11 20 12. Hãy vẽ cây kết quả.
- (b) Vẽ cây thu được ở câu a sau khi xóa gốc.

Câu 17. Sử dụng cây tìm kiếm nhị phân, hãy đưa ra thuật toán sắp xếp mảng theo thứ tự khóa tăng dần, bằng cách sử dụng các phép toán insert và deleteMin.

Câu 18. Giả sử tập dữ liệu được lưu giữ dưới dạng cây tìm kiếm nhị phân. Bài toán tìm kiếm phạm vi được xác định như sau: Cho hai giá trị khóa $k_1 < k_2$, ta cần tìm tất cả các dữ liệu d mà $k_1 \leq d.key \leq k_2$. Hãy thiết kế thuật toán cho bài toán tìm kiếm phạm vi.

Câu 19. Lớp ArrayStack biểu diễn ngăn xếp số nguyên cài bằng mảng cấp phát động. Hãy cài đặt phương thức `void push(int x);` thêm x vào đỉnh ngăn xếp. Chú ý cấp phát thêm bộ nhớ động khi mảng đầy.

```
class ArrayStack{
public:
    ArrayStack(): element(NULL), capacity(0), t(-1){}
    // Các phương thức khác của ngăn xếp
    void push(int x);
private:
    int * element; // con trỏ tới mảng cấp phát động
    int capacity; // dung lượng mảng động
}
```

```
int t; // chỉ số mảng của phần tử định ngăn xếp  
};
```

Câu 20. Lớp SList biểu diễn danh sách các số nguyên cài bằng danh sách liên kết đơn (DSLKD). Mỗi phần tử của DSLKD có kiểu Node định nghĩa bên dưới. SList có một thành viên dữ liệu duy nhất là head lưu địa chỉ phần tử đầu tiên của DSLKD.

Hãy định nghĩa hàm thành viên **appendList** nối một danh sách (truyền qua tham số someList) vào đuôi danh sách đang xét.

```
struct Node{  
    int data;  
    Node * next;  
};  
  
class SList{  
public:  
    SList(): head(NULL) {}  
    // Các phương thức khác của danh sách  
    void appendList(const SList & someList);  
private:  
    Node * head; // con trỏ tới phần tử đầu tiên của DSLKD  
};
```