

Môn thi: Cấu trúc dữ liệu và giải thuật
Mã lớp: Các lớp IT003 – Hệ đại trà, chất lượng cao
Thời gian làm bài: 90 phút
(Sinh viên không được sử dụng tài liệu)

Câu 1: (2 điểm)

- a. Hãy trình bày các bước của giải thuật sắp xếp chọn trực tiếp (Selection Sort) để sắp xếp một dãy số nguyên giảm dần (không viết hàm).
- b. Cho dãy số nguyên A như sau: 2, 5, 4, 7, 3, 9, 1, 8. Hãy cho biết dãy số A sẽ biến đổi như thế nào qua từng bước theo giải thuật ở câu 1a khi sắp xếp dãy số A giảm dần.



Đáp án tham khảo:

- a. Bước 1: Khởi tạo $i = 0$;
Bước 2: Một vòng lặp duyệt từ $arr[i]$ đến $arr[n]$ để tìm phần tử $arr[max_index]$ lớn nhất trong dãy.
Bước 3: Đổi chỗ $arr[max_index]$ và $arr[i]$
Bước 4: Nếu $i < n - 1$ thì
{
 $i = i + 1$;
 lặp lại bước 2;
}
ngược lại kết thúc thuật toán.

- b. Mảng đã cho gồm $n = 8$ phần tử:

2	5	4	7	3	9	1	8
---	---	---	---	---	---	---	---

Bước 1: $i = 0 \rightarrow arr[0] = 2$, $max_index = 5 \rightarrow$ đổi chỗ $arr[5]$ và $arr[0]$

9	5	4	7	3	2	1	8
---	---	---	---	---	---	---	---

Bước 2: $i = 1 \rightarrow arr[1] = 5$, $max_index = 7 \rightarrow$ đổi chỗ $arr[7]$ và $arr[1]$

9	8	4	7	3	2	1	5
---	---	---	---	---	---	---	---

Bước 3: $i = 2 \rightarrow arr[2] = 4$, $max_index = 3 \rightarrow$ đổi chỗ $arr[3]$ và $arr[2]$

9	8	7	4	3	2	1	5
---	---	---	---	---	---	---	---

Bước 4: $i = 3 \rightarrow arr[3] = 4$, $max_index = 7 \rightarrow$ đổi chỗ $arr[7]$ và $arr[3]$

9	8	7	5	3	2	1	4
---	---	---	---	---	---	---	---

Bước 5: $i = 4 \rightarrow arr[4] = 3$, $max_index = 7 \rightarrow$ đổi chỗ $arr[7]$ và $arr[4]$

9	8	7	5	4	2	1	3
---	---	---	---	---	---	---	---

Bước 6: $i = 5 \rightarrow arr[5] = 2$, $max_index = 7 \rightarrow$ đổi chỗ $arr[7]$ và $arr[5]$

9	8	7	5	4	3	1	2
---	---	---	---	---	---	---	---

Bước 7: $i = 6 \rightarrow arr[6] = 1$, $max_index = 7 \rightarrow$ đổi chỗ $arr[7]$ và $arr[6]$

9	8	7	5	4	3	2	1
---	---	---	---	---	---	---	---

Bước 8: $i = 7 = n - 1$ (không thỏa điều kiện $i < n - 1$); kết thúc giải thuật và ta được kết quả của dãy số sau khi sắp xếp giảm dần.

9	8	7	5	4	3	2	1
---	---	---	---	---	---	---	---

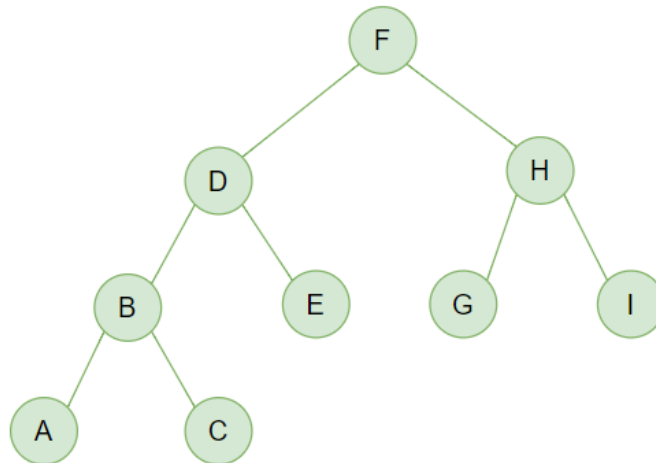
Câu 2: (4 điểm) Cho dãy ký tự như sau: F, D, H, B, A, G, C, E, I

Hãy thực hiện các yêu cầu sau:

- Vẽ cây nhị phân tìm kiếm bằng cách thêm lần lượt từng ký tự vào cây theo thứ tự từ trái qua phải của dãy ký tự trên, biết rằng giá trị của từng ký tự tương ứng theo thứ tự xuất hiện của ký tự trong từ điển (thứ tự Alphabet).
- Cho biết kết quả duyệt cây theo thứ tự: RNL và NRL.
- Hủy lần lượt từng nút theo thứ tự: D, E, F, H. Mỗi lần hủy 1 nút hãy vẽ lại cây.
- Viết hàm đếm số lượng nút lá có trên cây.

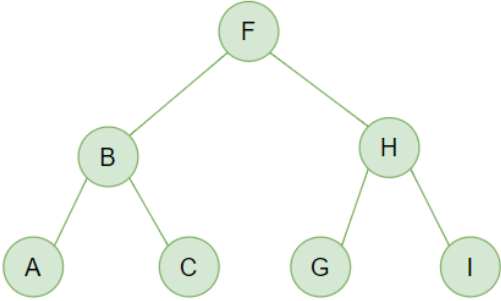
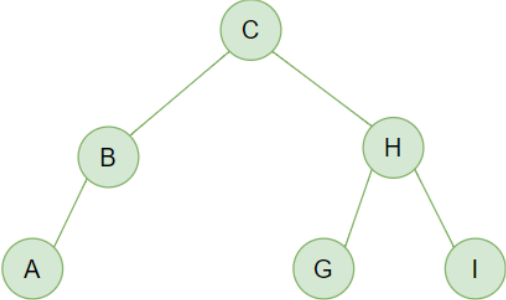
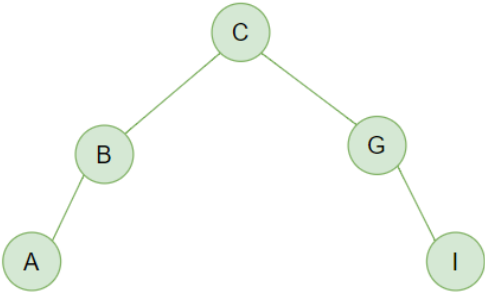
Đáp án tham khảo:

- Cây nhị phân tìm kiếm là cây nhị phân đảm bảo nguyên tắc bố trí mỗi khóa, mỗi nút sao cho:
 - Các nút của cây con bên trái nhỏ hơn nút hiện hành.
 - Các nút của cây con bên phải lớn hơn nút hiện hành.
 Để vẽ được cây nhị phân tìm kiếm bằng cái kí tự trên, ta phải biết thứ tự (giá trị) tương ứng của từng ký tự trong bảng chữ cái: $A < B < C < D < E < F < G < H < I$



- Duyệt cây theo RNL: I, H, G, F, E, D, C, B, A
Duyệt cây theo NRL: F, H, I, G, D, E, B, C, A
- Xóa D, E, F, H

Nút cần xóa	Dạng nút	Vẽ lại cây
D	nút có đầy đủ 2 cây con (chọn nút thế mạng là nút E)	<pre> graph TD F((F)) --> E((E)) F --> H((H)) E --> B((B)) E --> B --> A((A)) B --> C((C)) H --> G((G)) H --> I((I)) </pre>

E	nút có 1 cây con	 <pre> graph TD F((F)) --- B((B)) F --- H((H)) B --- A((A)) B --- C((C)) H --- G((G)) H --- I((I)) </pre>
F	nút có đầy đủ 2 cây con (chọn nút thể mang là nút C)	 <pre> graph TD C((C)) --- B((B)) C --- H((H)) B --- A((A)) H --- G((G)) H --- I((I)) </pre>
H	nút có đầy đủ 2 cây con (chọn nút thể mang là nút G)	 <pre> graph TD C((C)) --- B((B)) C --- G((G)) B --- A((A)) G --- I((I)) </pre>

```

d. int DemNutLa(TREE t) {
    if (t != NULL) {
        if (t->pLeft == NULL && t->pRight == NULL)
            count++;
        count += DemNutLa(t->pRight) + DemNutLa(t->pLeft)
    }
    return count;
}

```

Câu 3: (2 điểm) Cho cấu trúc dữ liệu lưu trữ thông tin nhân sự như sau:

```

typedef struct Thongtin {
    int maso; // mã số nhân sự
    char hoten[100]; // họ và tên nhân sự
    int thamnien; // số thâm niên công tác
    float hesoluong; // hệ số lương
} Nhansu;

```

Dựa trên một trong các thuật toán sắp xếp và tìm kiếm, hãy thực hiện yêu cầu sau:

- Viết hàm sắp xếp mảng NS gồm N nhân sự theo **thâm niên** công tác **giảm dần**:
void Sapxep_Thamnien(Nhansu NS[], int N) {...}
- Viết hàm để tìm xem trong một mảng NS gồm N nhân sự có nhân sự nào có mã số bằng X hay không. Nếu tìm thấy trả về giá trị 1, ngược lại trả về giá trị 0:
int Tim_Nhansu(Nhansu NS[], int N, int X) {...}

 **Đáp án tham khảo:**

```
a. void Sapxep_Thamnien(Nhansu NS[], int N) { //sắp xếp chọn selection sort
    int max_index;
    for (int i = 0; i < N - 1; i++) {
        max_index = i;
        for (int j = i + 1; j < N - 1; j++) {
            if (NS[j].thamnien > NS[max_index].thamnien)
                max_index = j;
        }
        swap(NS[max_index].thamnien, NS[i].thamnien)
    }
    void swap(Nhansu *xp, Nhansu *yp) {
        Nhansu temp = *xp;
        *xp = *yp;
        *yp = temp;
    }
b. int Tim_Nhansu(Nhansu NS[], int N, int X) {
    for (int i = 0; i < N; i++) {
        if (NS[i].maso == X)
            return 1;
        return 0;
    }
}
```

Câu 4: (2 điểm)

Cho K là một tập các giá trị khóa là số nguyên như sau: $K = \{89, 18, 10, 12, 49, 58, 69\}$ và bảng băm gồm $M = 10$ ô nhớ trống.

Cho 2 hàm băm như sau: $h_1(key) = key \bmod 10$ và $h_2(key) = (key \bmod 7) + 1$, trong đó phép toán *mod* là phép toán lấy phần dư.

Hãy vẽ hình bảng băm khi thêm lần lượt các khóa trong K vào bảng băm theo thứ tự từ trái qua phải bằng cách dùng hàm băm h_1 để xác định địa chỉ của mỗi khóa. Trong trường hợp xảy ra đụng độ thì dùng phương pháp băm kép (Double Hashing) để giải quyết đụng độ với hàm băm $h_i(key) = (h_1(key) + i * h_2(key)) \bmod 10$, trong đó $i = 1, 2, 3, \dots$ là số lần xảy ra đụng độ của khóa key ở lần thứ 1, 2, 3.

 **Đáp án tham khảo:**

Thêm 89: $h_1(89) = 89 \bmod 10 = 9$. Thêm 89 vào vị trí 9 trong bảng băm.

	Key
0	
1	
2	

3	
4	
5	
6	
7	
8	
9	89

Thêm 18: $h_1(18) = 18 \bmod 10 = 8$. Thêm 18 vào vị trí 8 trong bảng băm.

	Key
0	
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Thêm 10: $h_1(10) = 10 \bmod 10 = 0$. Thêm 10 vào vị trí 0 trong bảng băm.

	Key
0	10
1	
2	
3	
4	
5	
6	
7	
8	18
9	89

Thêm 12: $h_1(12) = 12 \bmod 10 = 2$. Thêm 12 vào vị trí 2 trong bảng băm.

	Key
0	10
1	
2	12
3	
4	
5	
6	
7	
8	18
9	89

Thêm 49: $h_1(49) = 49 \bmod 10 = 9$. Vị trí 9 chứa '49' \rightarrow xảy ra đụng độ

$$h_2(49) = (49 \bmod 7) + 1 = 1$$

- Băm lại lần 1: $(9 + 1.1) \bmod 10 = 0$. Vị trí 0 chứa '10' \rightarrow xảy ra đụng độ
- Băm lại lần 2: $(9 + 2.1) \bmod 10 = 1$. Thêm 49 vào vị trí 1.

	Key
0	10
1	49
2	12
3	
4	
5	
6	
7	
8	18
9	89

Thêm 58: $h_1(58) = 58 \bmod 10 = 8$. Vị trí 8 chứa '18' \rightarrow xảy ra đụng độ

$$h_2(58) = (58 \bmod 7) + 1 = 3$$

- Băm lại lần 1: $(8 + 1.3) \bmod 10 = 1$. Vị trí 1 chứa '49' \rightarrow xảy ra đụng độ
- Băm lại lần 2: $(8 + 2.3) \bmod 10 = 4$. Thêm 58 vào vị trí 4.

	Key
0	10
1	49
2	12
3	
4	58
5	
6	
7	
8	18
9	89

Thêm 69: $h_1(69) = 69 \bmod 10 = 9$. Vị trí 9 chứa '89' \rightarrow xảy ra đụng độ

$$h_2(69) = (69 \bmod 7) + 1 = 7$$

- Băm lại lần 1: $(9 + 1.7) \bmod 10 = 6$. Thêm 69 vào vị trí 6.

	Key
0	10
1	49
2	12
3	
4	58
5	
6	69
7	
8	18
9	89

HẾT