TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ THI HỌC PHẦN

Học kỳ: II Năm học: 2021 – 2022

Hệ đào tạo: Chính quy Bậc học: Đại học

Tên học phần: Cấu trúc dữ liệu và thuật toán

Số tín chỉ: 3

Ngày thi: 12/12/2022

Thời gian làm bài: 90 phút (Không kể thời gian phát đề)

Đề số: 4

Câu 1 (2 điểm) Bảng băm

1.1 . (1 điểm) Cho dãy giá trị (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) và bảng băm x mod 13, các nhận xét nào sau đây là đúng (giải thích lí do).

a. 9679, 1989, 4199 băm cùng ra 1 dãy giá trị
b. 1471, 6171 băm ra cùng một giá trị
c. Mọi phần tử đều băm ra cùng một giá trị
d. Các giá trị băm ra các giá trị khác nhau
e. Khác

1.2 . (1 điểm) Hàm băm nào giúp phân bố các giá trị băm đồng đều nhất vào các khóa có giá trị từ 0 tới 9 cho các giá trị x trong khoảng 0 tới 2020? Giải thích lí do.

a. h(i) = (12 * i + 3) mod 10
b. h(i) = 13 * i * i mod 10
c. h(i) = (i * i * i + 7) mod 10
d. h(i) = (i * i + 9) mod 10

Câu 2 (2 điểm) Thuật toán sắp xếp chọn (Selection Sort).

2.1 (1 điểm) Trình bày ý tưởng của thuật toán sắp xếp chọn.

2.2 (1 điểm) Cho đoạn code của sắp xếp chọn trên danh sách liên kết như sau:

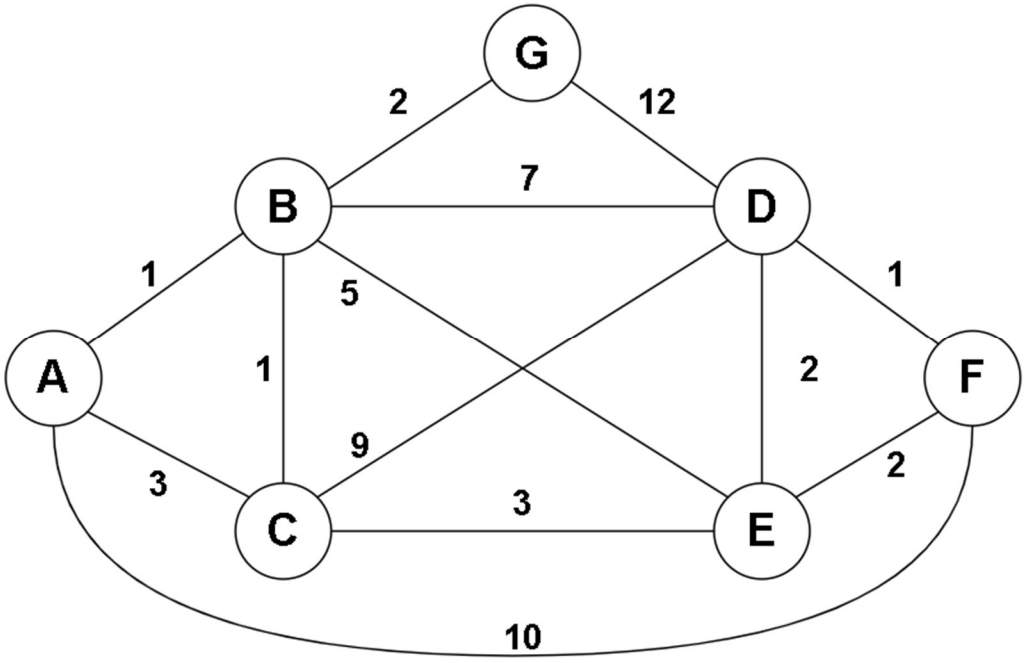
```
void selectionSort(node* head)
{
	node* temp = head;
	// Traverse the List
	while (temp) {
		node* min = temp;
		node* r = temp->next;

		// Traverse the unsorted sublist
		while (r) {
			if (min->data>r->data)
				min = r;

			r=r->next;
		}
		// Swap Data
		int x = temp->data;
		temp data = min->data;
		min->data = x;
		temp temp->next;
	}
}
```

Cho một dãy số được biểu diễn bởi danh sách liên kết đơn gồm các số 7, 3, 4, 6, 1, 12, 10, 5, 9, 8, 2 theo đúng thứ tự. Hãy vẽ danh sách liên kết ở lần lặp thứ 4 và thứ 7 dựa vào đoạn code cung cấp bên trên.

Câu 3. (2 điểm) Cho đồ thị sau:



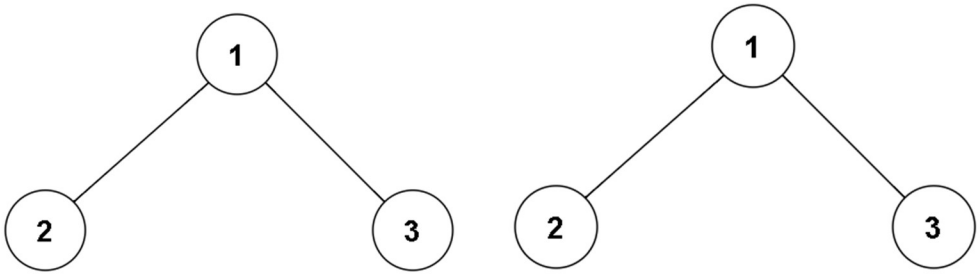
3.1 (1 điểm) Minh họa cách biểu diễn đồ thị trên bằng ma trận kề.

3.2 (1 điểm) Minh họa cách thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh A đến đỉnh D.

Câu 4 (2 điểm) Cây nhị phân

Hai cây có cùng nội dung nếu chúng tại mọi nút đều có cùng số lượng con trái, con phải và các nút tương ứng cùng giá trị.

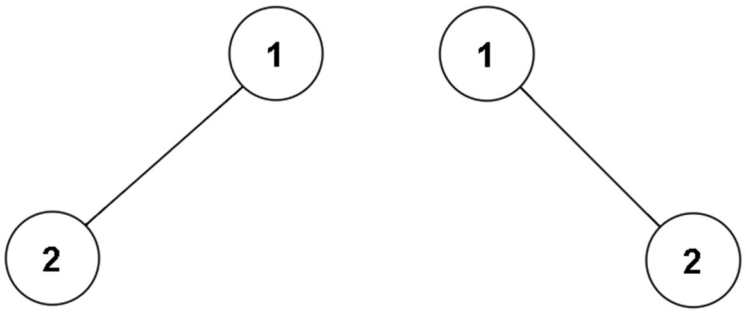
Ví dụ 1:



Dữ liệu vào: p = [1,2,3], q = [1,2,3]

Dữ liệu ra: true

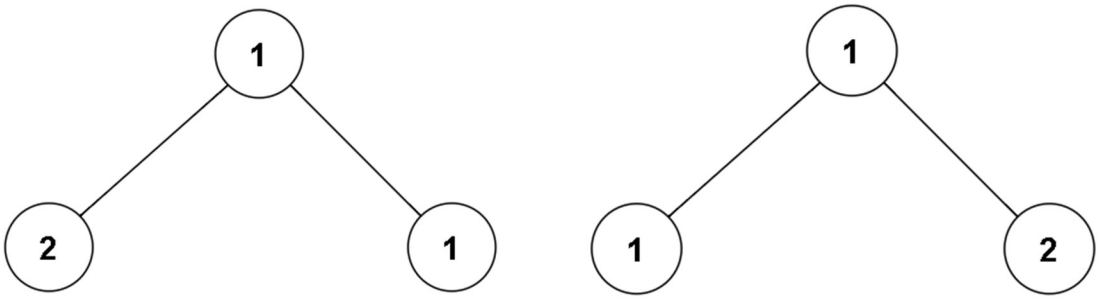
Ví dụ 2:



Dữ liệu vào: p = [1,2], q = [1,null,2]

Dữ liệu ra: false (vì cây thứ hai không có con trái của nút gốc)

Ví dụ 3:



Dữ liệu vào: p = [1,2,1], q = [1,1,2]

Dữ liệu ra: false (vì con trái của nút gốc 1 ở cây bên trái có giá trị là 2, trong khi con trái của nút gốc 1 ở cây bên phải có giá trị là 1)

Trong máy tính, cây có thể biểu diễn bằng cấu trúc node như sau:

```
// Cấu trúc của các node trong cây tìm kiếm nhị phân
```

```
Struct TreeNode {
        int val;
        struct TreeNode *left;
        struct TreeNode *right;
};
```

4.1 (1 điểm) Nêu mã giả của việc kiểm tra hai cây có cùng nội dung hay không.

4.2, (1 điểm) Cho đoạn code sau, sắp xếp các câu lệnh để được đoạn code dùng thực hiện việc kiểm tra hai cây nhị phân có cùng nội dung

```
1. bool result;
2.
3. void order(struct TreeNode p, struct TreeNode* q)
4. {
5.      if(result= false) return;
6.      order(p->left,q->left);
7.      order(pright.q-right).
8.
9.      if(p-NULL | q-NULL)
10.     {
11.             result-false;
12.             return;
13.
14.             if(p-NULL &&q==NULL)
15.             {
16.                     return;
17.             }
```

```
18.             else
19.             {
20.                     return;
21.             }
22.     }
23.
24.     result-false;
25.
26.     if(p->val!-q->val)
27.     {
28.             return;
29.     }
30.
31. }
32.
33. bool isSameTree(struct TreeNode" p, struct TreeNode* q) {
34.     order(p,q);
35.     result=true;
36.     return result;
37. }
```

Câu 5. (2 điểm) Tìm kiếm

Cho một dãy số nguyên A[1..n]. Cho số nguyên X, kiểm tra X có nằm trong dãy số A hay không.

5.1. (1 điểm) Nếu giá trị các dãy số đều nằm trong đoạn 1 .. 1,000,000; cho phép sử dụng bộ nhớ phụ, ta có thể lưu trữ như nào để việc tìm kiếm số nguyên X bất kỳ có thể được thực hiện với độ phức tạp O(1).

5.2. (1 điểm) Giả thiết A[i] nhận giá trị bất kỳ và mảng A đã được sắp xếp giảm. Viết thuật toán tìm kiếm nhị phân hàm đếm số phần tử bằng X trong dãy A[1..n] với độ phức tạp O(logn).

```
// hàm trả về số phần tử bằng phần tử X trong mảng A với độ phức tạp O(logn)

int logCount(int A[], int N, int X){
    // bổ sung code cho hàm này
}
```