

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

Câu 1:

- Hãy cho biết độ phức tạp của thuật toán Insertion sort (chèn trực tiếp) theo định nghĩa Big-O ( $O$  lớn) (0.25 điểm)
- Viết hàm sắp xếp mảng 1 chiều gồm  $N$  phần tử giảm dần với thuật toán Insertion sort (0.75 điểm)
- Hãy cho biết dãy số sẽ thay đổi qua từng bước như thế nào khi áp dụng thuật toán ở câu 1b, biết rằng dãy số cho như sau: 3, 8, 4, 5, 9, 1, 2, 6 (1 điểm)

Câu 2:

Cho dãy ký tự như sau: R, E, T, A, V, X, L, G, S, I

Hãy thực hiện các yêu cầu sau:

- Vẽ cây nhị phân 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 (1 điểm)
- Cho biết kết quả duyệt cây theo RNL, NRL (1 điểm)
- Huỷ lần lượt từng nút L, T, E, R trên cây, mỗi lần huỷ 1 nút vẽ lại cây nối tiếp theo như thứ tự huỷ (1 điểm)

Câu 3:

Cho biết cây B-Tree bậc 3 là một cây thỏa mãn các tính chất sau:

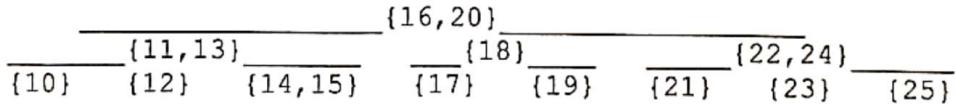
- Tất cả node lá nằm trên cùng một mức
- Tất cả các node, trừ node gốc và node lá, có \*tối thiểu\* 2 node con.
- Tất cả các node có \*tối đa\* 3 con
- Tất cả các node, trừ node gốc, có từ 1 cho đến 2 khóa (keys)
- Một node không phải lá và có  $n$  khóa thì phải có  $n+1$  node con.

Hãy thực hiện các yêu cầu sau:

3.1 Cho dãy số: 12, 17, 20, 23, 15, 11, 24, 13, 19, 22, 18, 21, 16. Hỏi khi lần lượt thêm các số trong dãy theo thứ tự từ trái qua phải vào một cây B-Tree bậc 3 rỗng thì:

- Các khóa nào khi thêm vào cây sẽ làm phát sinh thao tác tách (split) node? (0.5 điểm)
- Vẽ cây B-Tree trước và sau khi thêm các khóa ở câu a (1 điểm)

3.2 Cho cây B-Tree bậc 3 như hình sau:



Hãy lần lượt tiến hành xóa các khóa sau khỏi cây: 13, 24, 19 và vẽ cây B-Tree trước và sau khi xóa mỗi khóa trên (0.5 điểm)

Lưu ý khi xoá:

- Khi khóa cần xóa (gọi là  $x$ ) không nằm ở node lá, chọn khóa thế mạng là khóa có giá trị lớn nhất mà nhỏ hơn  $x$ .

- Thao tác nhường khoá (underflow) sẽ được thực hiện khi hai node liền kề có tổng số khoá  $\geq 2$ . Khi có một node không còn đáp ứng dù số lượng khoá tối thiểu, ưu tiên thực hiện underflow thay cho catenation (hợp) vì thao tác này không làm thay đổi số khoá của node cha.
- Khi có 02 lựa chọn node liền kề để thực hiện catenation, ưu tiên chọn catenation giữa node bị thiếu khoá với node liền trước.

#### Câu 4:

Để việc tìm kiếm thông tin mặt hàng được nhanh chóng, người ta dùng một bảng băm theo phương pháp thăm dò, làm việc trên mã quản lý của mặt hàng. Mã quản lý này là một con số nguyên. Bảng băm có:

- Hỗn băm:  $h(key) = (key \% M)$
- Hỗn băm lại (hỗn thăm dò):  $prob(key, i) = (h(key) + i*i + i) \% M$

Trong đó:

- key là giá trị khóa.
- i là một số nguyên cho biết lần băm lại (thăm dò) thứ i.
- M là kích thước bảng băm.

Giả sử  $M = 7$ , cho trường hợp  $T$  của bảng băm đã chứa dữ liệu như bên dưới. Biết “-” là ký hiệu vị trí trống trong bảng băm.

Bảng băm T	
0	-
1	-
2	16
3	-
4	-
5	12
6	13

- a. Trình bày từng bước việc tìm mã quản lý 23 trong bảng băm  $T$ . (0.5 điểm)
- b. Trình bày từng bước việc thêm các mã quản lý sau vào bảng băm  $T$  theo đúng thứ tự liệt kê là 11, 20, 27 (1.5 điểm).

#### Câu 5:

Trong các ứng dụng thực tế, chẳng hạn trong mạng lưới giao thông đường bộ, đường thủy hoặc đường hàng không, người ta không chỉ quan tâm đến việc tìm đường đi giữa hai địa điểm mà còn phải lựa chọn một hành trình tiết kiệm nhất (theo tiêu chuẩn không gian, thời gian hay chi phí). Vấn đề này có thể được mô hình hóa thành một bài toán trên đồ thị, trong đó mỗi địa điểm được biểu diễn bởi một đỉnh, cạnh nối hai đỉnh biểu diễn cho “đường đi trực tiếp” giữa hai địa điểm (tức không đi qua địa điểm trung gian) và trọng số của cạnh là khoảng cách giữa hai địa điểm.

Bài toán có thể phát biểu dưới dạng tổng quát như sau: Cho một đơn đồ thị có hướng và có trọng số dương  $G=(V,E)$ , trong đó  $V$  là tập đỉnh,  $E$  là tập cạnh (cung) và các cạnh đều có trọng số, hãy tìm một đường đi (không có đỉnh lặp lại) ngắn nhất từ đỉnh xuất phát  $S$  thuộc  $V$  đến đỉnh đích  $F$  thuộc  $V$ .

Giả sử thông tin đầu vào của bài toán (Input) được nhập vào chương trình như sau:

Input	Giải thích
7	- Dòng đầu tiên chứa một số nguyên dương $e$ cho biết số cạnh của đồ thị
A B 1	- Với $e$ dòng tiếp theo, mỗi dòng chứa hai chuỗi $u, i$ và một số nguyên
B E 3	đương $x$ , thể hiện thông tin có một cạnh nối từ đỉnh $u$ sang đỉnh $i$ trong

E D 3	đồ thị với độ dài (trọng số) là x
C B 4	- Dòng cuối cùng chứa hai chuỗi s và f, đây là đỉnh bắt đầu và đỉnh kết thúc của đường đi cần tìm
A D 7	
E C 2	
C D 1	<u>Lưu ý: không biết trước số đỉnh và danh sách các đỉnh.</u>
A E	

Hãy thực hiện các yêu cầu sau:

- Xây dựng các cấu trúc dữ liệu phù hợp nhất có thể để biểu diễn đồ thị trên máy tính theo input đã cho. (0.5 điểm)  
Cấu trúc được xem là tốt nếu đạt được các tiêu chuẩn sau: Tiết kiệm tài nguyên; Hỗ trợ một số thao tác cơ bản như “Kiểm tra hai đỉnh có kề nhau không”, “Tim danh sách các đỉnh kề với một đỉnh cho trước” với ràng buộc là không phải duyệt qua danh sách tất cả các cạnh của đồ thị.
- Viết hàm nhập theo Input ở đầu bài và lưu trữ thông tin của đồ thị vào cấu trúc dữ liệu đã đề xuất ở câu a. (0.5 điểm)

\*\*\* KHÔNG YÊU CẦU tìm cách giải cho bài toán này. Sinh viên ĐƯỢC PHÉP sử dụng Standard Template Library-STL với những cấu trúc dữ liệu (vector, stack, queue, list, map, set, pair, ...) cũng như giải thuật được xây dựng sẵn.

Hết

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)

- 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).
- 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 1.a khi sắp xếp dãy số A giảm dần.

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.

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 năm 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) {...}

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 khoá. 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  $i=1,2,3..$  là số lần xảy ra đụng độ của khóa key ở lần thứ 1,2,3..

HẾT

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). (1 điểm)

Đáp án tham khảo và Thang điểm đề nghị :

Bước 1: Khởi gán $i = 0$ ; // vị trí bắt đầu dãy N phần tử	0.25 điểm
Bước 2: Tìm phần tử $a[\max]$ lớn nhất trong dãy số hiện hành từ $a[i]$ đến $a[N]$	0.25 điểm
Bước 3 : Đổi chỗ $a[\max]$ và $a[i]$	0.25 điểm
Bước 4 : Nếu $i < N-1$ thì { $i=i+1$ ; Lặp lại Bước 2; } Ngược lại: Dừng giải thuật.	0.25 điểm

- 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 1.a khi sắp xếp dãy số A giảm dần. (1 điểm)

Đáp án tham khảo và Thang điểm đề nghị :

Dãy số ban đầu :

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

- Lần lặp 1 và dãy số biến đổi: 0.125 điểm

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

- Lần lặp 2 và dãy số biến đổi: 0.125 điểm

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

- Lần lặp 3 và dãy số biến đổi: 0.125 điểm

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

- Lần lặp 4 và dãy số biến đổi: 0.125 điểm

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

- Lần lặp 5 và dãy số biến đổi: 0.125 điểm

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

- Lần lặp 6 và dãy số biến đổi: 0.125 điểm

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

- Lần lặp 7 và dãy số biến đổi: 0.125 điểm

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

- Kết thúc giải thuật, trả lời dãy số kết quả : 0.125 điểm

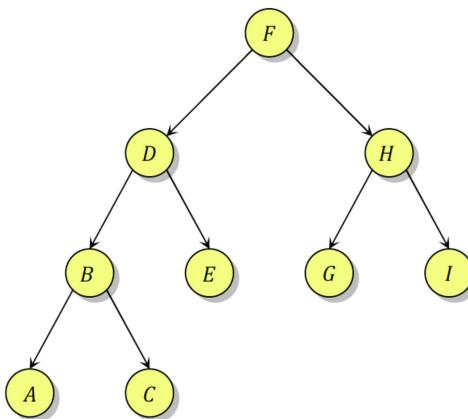
9	8	7	5	4	3	2	1
---	---	---	---	---	---	---	---

**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 :

- a. 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) (1 điểm)

Đáp án :



Thang điểm đề nghị :

- Vẽ cây chính xác : 1 điểm
- Vẽ sai 1 nút trên cây : 0 điểm

- b. Cho biết kết quả duyệt cây theo thứ tự: RNL và NRL. (1 điểm)

Đáp án và Thang điểm đề nghị :

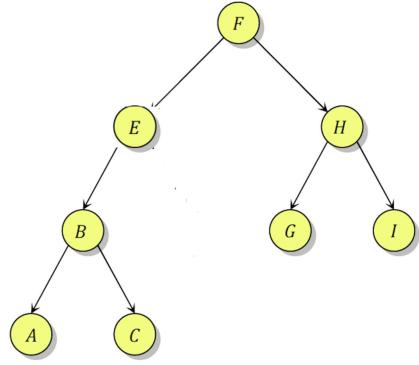
- Duyệt cây RNL : I,H,G,F,E,D,C,B,A (0.5 điểm)

- Duyệt cây NRL : F,H,I,G,D,E,B,C,A (0.5 điểm)

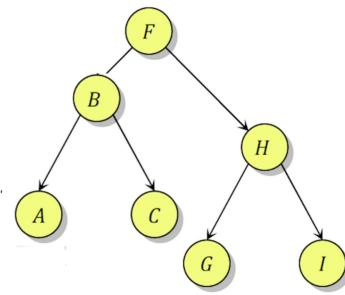
- c. Huỷ lần lượt từng nút theo thứ tự: D, E, F, H. Mỗi lần huỷ 1 nút hãy vẽ lại cây. (1 điểm)

Đáp án tham khảo :

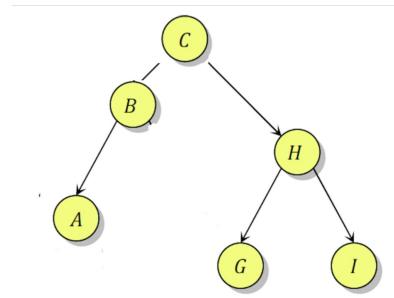
- Xoá nút D :



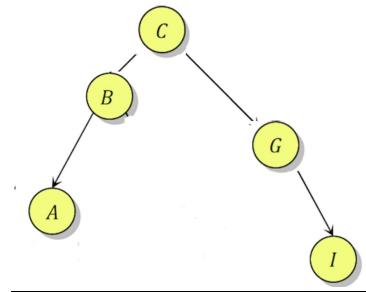
- Xoá nút E :



- Xoá nút F :



- Xoá nút H :



Thang điểm đề nghị :

- Thực hiện Xoá chính xác một nút : 0.25 điểm / 1 nút

- Trường hợp SV không xoá lần lượt theo yêu cầu của đề, chỉ vẽ cây kết quả cuối cùng sau khi xoá cả 4 nút và kết quả vẽ cây là đúng, nhưng không có giải thích vì sao có được kết quả: trừ 50% số điểm của câu.
- d. Viết hàm đếm số lượng nút lá có trên cây. (1 điểm)

Đáp án tham khảo và Thang điểm đề nghị :

<pre>int DEMLA(Tree T) {     if (T==NULL)         return 0;     if (T-&gt;Left==NULL &amp;&amp; T-&gt;Right==NULL)         return 1;     else         return DEMLA(T-&gt;Left) + DEMLA(T-&gt;Right); }</pre>	0.25 điểm 0.25 điểm 0.5 điểm
--	------------------------------------

**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 năm 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:

- a. Viết hàm sắp xếp mảng NS gồm N nhân sự theo **thâm niên** công tác **giảm dần** (1 điểm)

Đáp án tham khảo và Thang điểm đề nghị :

- SV có thể sử dụng bất kỳ giải thuật sắp xếp nào để áp dụng khi viết hàm.
- Đáp án tham khảo giải thuật chọn trực tiếp và thang điểm đề nghị.

<pre>void Sapxep_Thamnien (Nhansu NS[], int N) {     int i, j, max_idx;     for (i = 0; i &lt; N-1; i++)     {         max_idx = i;         for (j = i+1; j &lt; N; j++)             if (NS[j].thamnien &gt; NS[max_idx].thamnien)                 max_idx = j;         swap(&amp;NS[max_idx], &amp;NS[i]);     } }</pre>	0.25 điểm 0.25 điểm 0.25 điểm 0.25 điểm
---	--

```
void swap(Nhansu *xp, Nhansu *yp)
{
```

```

Nhansu temp = *xp;
*xp = *yp;
*yp = temp;
}

```

- b. Viết hàm để tìm xem trong một mảng NS gồm N nhân số có nhân số nào có mã số bằng X hay không. Nếu tìm thấy trả về giá trị 1, ngược lại trả về giá trị 0 (1 điểm)

Dáp án tham khảo và Thang điểm đề nghị :

- SV có thể sử dụng bất kỳ giải thuật tìm kiếm nào để áp dụng khi viết hàm.
- Đáp án tham khảo giải thuật tìm kiếm tuyến tính và thang điểm đề nghị.

int Tim_Nhansu (Nhansu NS[], int N, int X)	0.25 điểm
{ int i;	
for(i=0; i<N; i=i+1)	0.25 điểm
if (NS[i].maso == X)	0.25 điểm
return 1;	
return 0;	0.25 điểm
}	

#### 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  $\bmod$  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 khoá. 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  $i=1,2,3..$  là số lần xảy ra đụng độ của khóa key ở lần thứ 1,2,3..

Dáp án tham khảo và Thang điểm đề nghị :

- Các số khi thêm vào sử dụng hàm  $h_1(key)$  : 89, 18, 10, 12, không bị đụng độ.
- Thang điểm đề nghị : 0.25 điểm / 1 số thêm vào bảng băm đúng.

- Bảng băm khi thêm 89 :  
+ $h_1(89) = 9$

Vị trí	Giá trị
0	
1	
2	
3	
4	

- Bảng băm khi thêm 18 :  
+ $h_1(18)=8$

Vị trí	Giá trị
0	
1	
2	
3	
4	

- Bảng băm khi thêm 10 :  
+ $h_1(10)=0$

Vị trí	Giá trị
0	10
1	
2	
3	
4	

<u>5</u>	
<u>6</u>	
<u>7</u>	
<u>8</u>	
<u>9</u>	<u>89</u>

<u>5</u>	
<u>6</u>	
<u>7</u>	
<u>8</u>	<u>18</u>
<u>9</u>	<u>89</u>

<u>5</u>	
<u>6</u>	
<u>7</u>	
<u>8</u>	<u>18</u>
<u>9</u>	<u>89</u>

- Bảng băm khi thêm 12 :  
 $+h1(12)=2$

Vị trí	Giá trị
<u>0</u>	<u>10</u>
<u>1</u>	
<u>2</u>	<u>12</u>
<u>3</u>	
<u>4</u>	
<u>5</u>	
<u>6</u>	
<u>7</u>	
<u>8</u>	<u>18</u>
<u>9</u>	<u>89</u>

- Các số khi thêm vào sử dụng hàm  $h1(key)$ ,  $h2(key)$  : 49, 58, 69, bị đụng độ.

- Bảng băm khi thêm 49 : 0.5 điểm

$$+h1(49)=9$$

$$+h2(49) = (49 \% 7) + 1 = 1$$

$$+ \text{vị trí } h_1(49) = h1(49) + 1 * h2(49) = 9 + 1 * 1 = 10 \% 10 = 0 \text{ (đụng độ)}$$

$$+ \text{vị trí } h_2(49) = h1(49) + 2 * h2(49) = 9 + 2 * 1 = 11 \% 10 = 1 \text{ (chấp nhận)}$$

Vị trí	Giá trị
<u>0</u>	<u>10</u>
<u>1</u>	<u>49</u>
<u>2</u>	<u>12</u>
<u>3</u>	
<u>4</u>	
<u>5</u>	
<u>6</u>	
<u>7</u>	
<u>8</u>	<u>18</u>
<u>9</u>	<u>89</u>

- Bảng băm khi thêm 58 : 0.25 điểm

$$+h1(58)=8$$

$$+h2(58) = (58 \% 7) + 1 = 3$$

$$+ \text{vị trí } h_1(58) = h1(58) + 1 * h2(58) = 8 + 1 * 3 = 11 \% 10 = 1 \text{ (đụng độ)}$$

$$+ \text{vị trí } h_2(58) = h1(58) + 2 * h2(58) = 8 + 2 * 3 = 14 \% 10 = 4 \text{ (chấp nhận)}$$

Vị trí	Giá trị
0	10
1	49
2	12
3	
4	58
5	
6	
7	
8	18
9	89

- Bảng băm khi thêm 69 : 0.25 điểm

$$+h_1(69) = 9$$

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

$$+ \text{vị trí } h_1(69) = h_1(69) + 1 * h_2(69) = 9 + 1 * 7 = 16 \% 10 = 6 \text{ (chấp nhận)}$$

Vị trí	Giá trị
0	10
1	49
2	12
3	
4	58
5	
6	69
7	
8	18
9	89

- Trong trường hợp SV thêm các số 49,58,69 vào bảng băm chính xác, nhưng không có giải thích vì sao có kết quả, đề nghị điểm cho trường hợp này là 0.5 điểm.

*Bảng tóm tắt phân bố thang điểm :*

Câu	Thang điểm
1.a	1
1.b	1
2.a	1
2.b	1
2.c	1
2.d	1

3.a	1
3.b	1
4. Ý 1 : các số không đúng đắn	1
4. Ý 2 : các số bị đúng đắn	1
Tổng điểm	10

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: (2.5 điểm)**

- Trình bày các bước giải thuật sắp xếp Quick sort (không viết chương trình) để sắp xếp mảng số nguyên N phần tử **giảm dần**, cho biết độ phức tạp giải thuật.
- Trình bày các bước (vẽ từng bước) áp dụng giải thuật trong câu 1.a để sắp xếp mảng số nguyên {10, 5, 30, 70, 40, 80, 90} giảm dần.

**Câu 2: (4 điểm)**

Cho dãy số sau: 11, 6, 8, 19, 4, 10, 5, 17, 43, 49, 31

Hãy thực hiện các yêu cầu sau:

- Xây dựng **cây nhị phân tìm kiếm** từ dãy số đã cho vào cây theo thứ tự thêm các số từ trái sang phải của dãy số.
- Duyệt cây trong câu 2.a theo Node-Left-Right, Right-Left-Node.
- Xóa khỏi cây **lần lượt các nút** 8, 11, 43, 6 (vẽ hình từng trường hợp) sao cho cây vẫn là cây nhị phân tìm kiếm sau khi xóa nút.
- Viết hàm in ra màn hình các nút trên cây có duy nhất một nút con.

**Câu 3: (4 điểm)**

Cho bảng băm A kích thước 13 phần tử và tập khóa  $K = \{10, 26, 52, 76, 13, 8, 3, 33, 60, 42\}$ , ta cần nạp các giá trị khóa  $K$  vào bảng băm A sử dụng hàm băm  $H(K) = K \% 13$ .

Hãy vẽ bảng băm khi **thêm từng khóa K vào bảng A**, trong trường hợp xảy ra đụng độ, sử dụng phương pháp dò tuyển tính để giải quyết đụng độ.

**Câu 4: (1.5 điểm)**

Mạng xã hội là dịch vụ kết nối các thành viên, người dùng trên Internet lại với nhau dựa theo những tiêu chí, sở thích nào đó, với nhiều mục đích khác nhau, là nơi trao đổi thông tin, chia sẻ suy nghĩ, ý tưởng mà không bị giới hạn về không gian và thời gian. Các thành viên giao tiếp với các thành viên khác trong mạng, mỗi thành viên sẽ là một chủ thể trao đổi thông tin và tương tác với người khác.

Bạn được giao nhiệm vụ xây dựng một mạng xã hội thu nhỏ với các chức năng : kết bạn với nhau, xem thông tin của bạn của mình post trên mạng và like một post của bạn giống như cách trên Facebook, hãy đề xuất ý tưởng :

- Mô tả cấu trúc dữ liệu bạn nghĩ sẽ sử dụng phù hợp với các chức năng của mạng đã mô tả ở phần trên.
- Mô tả các giải thuật (các bước thực hiện giải thuật, không cài đặt) để hiện thực các chức năng của mạng đã mô tả ở phần trên và phù hợp với cấu trúc dữ liệu bạn đã đề nghị trong câu 4.a.

**HẾT**

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: (2.5 điểm)**

- a. Trình bày các bước giải thuật sắp xếp Quick sort (không viết chương trình) để sắp xếp mảng số nguyên N phần tử **giảm dần**, cho biết độ phức tạp giải thuật. (1.5 điểm)

**Thang điểm đề nghị :**

- SV viết các bước thuật toán sử dụng mã giả : full điểm
- SV trình bày được các bước chung của QuickSort : full điểm

B1: chọn 1 phần tử trong mảng làm pivot (phần tử bất kỳ / đầu mảng / cuối mảng / giữa mảng)

B2: phân hoạch dãy số làm 2 (or 3 phần), phần bên trái gồm những phần tử có giá trị lớn hơn pivot, phần bên phải thì giá trị nhỏ hơn.

B3: Gọi đệ quy QuickSort để sắp xếp 2 dãy con này.

- SV viết code đúng thay vì mã giả: full điểm

**Ý 2 : Trình bày độ phức tạp giải thuật : 0.5 điểm**

- SV viết chính xác độ phức tạp trung bình, tốt nhất : 0.25 điểm / 1 độ phức tạp viết đúng.

- b. Trình bày các bước (vẽ từng bước) áp dụng giải thuật trong câu 1.a để sắp xếp mảng số nguyên {10, 5, 30, 70, 40, 80, 90} giảm dần. (1 điểm)

Mảng ban đầu :

10	5	30	70	40	80	90
----	---	----	----	----	----	----

Lần 1 : Chọn phần tử 70 làm phần tử phân hoạch:

90	80	70	30	40	5	10
----	----	----	----	----	---	----

Lần 2 : Chọn phần tử 80 làm phần tử phân hoạch [đoạn 90,80]

90	80	70	30	40	5	10
----	----	----	----	----	---	----

Lần 3 : Chọn phần tử 5 làm phần tử phân hoạch [đoạn 30,40,5,10]

90	80	70	30	40	10	5
----	----	----	----	----	----	---

Lần 4 : Chọn phần tử 40 làm phần tử phân hoạch [đoạn 30,40,10]

90	80	70	40	30	10	5
----	----	----	----	----	----	---

Kết thúc : Mảng sắp giảm : 90, 80, 70, 40, 30, 10, 5

### Thang điểm đề nghị :

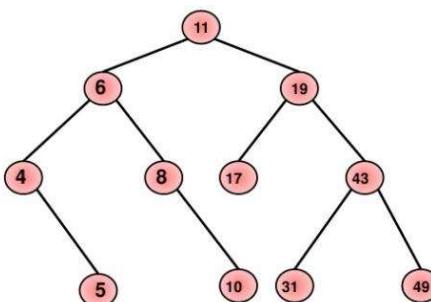
- SV phải ghi rõ chọn giá trị nào làm trục ở mỗi bước, swap đúng các cặp phần tử và sau bước phân hoạch thực hiện đệ quy QuickSort CHO NHỮNG ĐOẠN CỤ THỂ NÀO. Nếu không có chủ thích gì mà chỉ ghi sự thay đổi của mảng ở mỗi bước thì đạt tối đa 0.5 điểm.

### **Câu 2: (4 điểm)**

Cho dãy số sau: 11, 6, 8, 19, 4, 10, 5, 17, 43, 49, 31

Hãy thực hiện các yêu cầu sau:

- Xây dựng **cây nhị phân tìm kiếm** từ dãy số đã cho vào cây theo thứ tự thêm các số từ trái sang phải của dãy số (1 điểm)



### Thang điểm đề nghị :

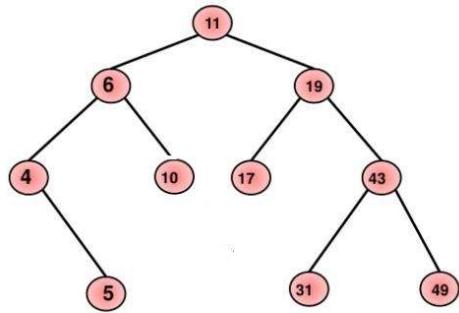
- Vẽ chính xác cây hoàn chỉnh : được trọn 1 điểm.
- Vẽ sai 1 số trên cây -> cây không còn là cây nhị phân tìm kiếm : không có điểm.
- Tạo cây với thứ tự sai của dãy số đã cho : không có điểm.

- Duyệt cây trong câu 2.a theo Node-Left-Right, Right-Left-Node (1 điểm)

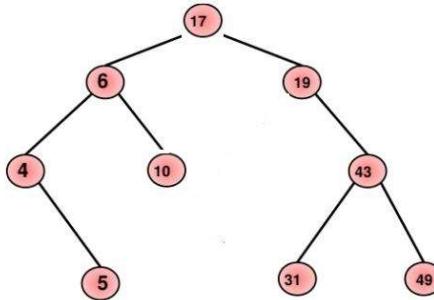
### Thang điểm đề nghị :

- Node – Left – Right viết đúng : 0,5 điểm – 11,6,4,5,8,10,19,17,43,31,49
- Right – Left - Node viết đúng : 0,5 điểm – 49,31,43,17,19,10,8,5,4,6,11

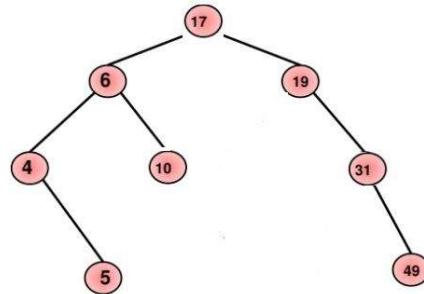
- Xóa khỏi cây **lần lượt các nút** 8, 11, 43, 6 (vẽ hình từng trường hợp) sao cho cây vẫn là cây nhị phân tìm kiếm sau khi xoá nút. (1 điểm)  
+ Xóa nút 8 :



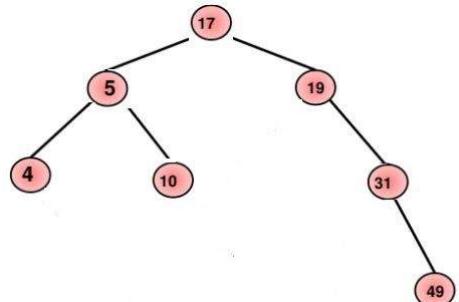
+ Xóa nút 11:



+ Xóa nút 43 :



+ Xóa nút 6 :



### Thang điểm đề nghị :

- Xóa từng số đúng và vẽ lại cây chính xác sau khi xóa số : 0,25 điểm / 1 số thực hiện đúng.
- Do câu hỏi là lý thuyết, không phải yêu cầu thực hiện dựa trên một hàm cài đặt cụ thể cho trước, nên yêu tố quan trọng nhất là làm đúng theo phương pháp đã

- học, do đó SV xoá cây bên trái / bên phải cũng nên được đồng ý full điểm của câu
- Nếu SV không ghi rõ từng bước xóa mà chỉ vẽ cây kết quả cuối cùng và đúng : 0.5 điểm

- d. Viết hàm in ra màn hình các nút trên cây có duy nhất một nút con (1 điểm)

```
void Print(Tree T)
```

```
{ if (T!=NULL)
```

```
{ if ((T->pLeft != NULL && T->pRight == NULL) || (T->pLeft == NULL  
&& T->pRight != NULL))
```

```
    cout << T->key;
```

```
    Print(T->pLeft);
```

```
    Print(T->pRight);
```

```
}
```

```
}
```

#### Thang điểm đề nghị :

- Hoàn thành chính xác hàm : được trọn 1 điểm.
- Thiếu 1 điều kiện (con trỏ trái của nút khác NULL và con trỏ phải là NULL) HOẶC (con trỏ trái của nút là NULL và con trỏ phải khác NULL) : trừ 0.5 điểm

#### **Câu 3: (2 điểm)**

Cho bảng băm A kích thước 13 phần tử và tập khóa K = {10, 26, 52, 76, 13, 8, 3, 33, 60, 42}, ta cần nạp các giá trị khóa K vào bảng băm A sử dụng hàm băm H(K) = K % 13.

Hãy vẽ bảng băm khi **thêm từng khóa K vào bảng A**, trong trường hợp xảy ra đụng độ, sử dụng phương pháp dò tuyển tính để giải quyết đụng độ.

Vị trí	Giá trị lưu	Đụng độ ?
12		
11	76	
10	10	
9	60	Đụng độ 60% 13 = 8
8	8	
7	33	
6		
5		
4	42	Đụng độ 42%13 = 3
3	3	
2	13	Đụng độ 13%13 = 0

1	52	Đụng độ 52% = 0
0	26	

### Thang điểm đề nghị :

- 4 số bị đụng độ 52, 13, 42, 60 đưa vào bảng băm chính xác : 0.25 điểm cho 1 số.
- 6 số không bị đụng độ 26, 3, 33, 8, 10, 76 đưa vào bảng băm chính xác : 0.125 điểm cho 1 số.
- Hoàn thành chính xác bảng băm : được trọn 2 điểm.
- SV phải ghi rõ được các trường hợp đụng độ thế nào và hướng xử lý. Nếu chỉ ghi bảng kết quả cuối cùng thì chỉ đạt tối đa 1.5 điểm.
- Trường hợp SV chỉ tính toán hàm băm mà không vẽ lại bảng băm (gồm bao nhiêu phần tử, giá trị tại mỗi phần tử) thì cũng trừ điểm 0.5.

### **Câu 4: (1.5 điểm)**

Mạng xã hội là dịch vụ kết nối các thành viên, người dùng trên Internet lại với nhau dựa theo những tiêu chí, sở thích nào đó, với nhiều mục đích khác nhau, là nơi trao đổi thông tin, chia sẻ suy nghĩ, ý tưởng mà không bị giới hạn về không gian và thời gian. Các thành viên giao tiếp với các thành viên khác trong mạng, mỗi thành viên sẽ là một chủ thể trao đổi thông tin và tương tác với người khác.

Bạn được giao nhiệm vụ xây dựng một mạng xã hội thu nhỏ với các chức năng : kết bạn với nhau, xem thông tin của bạn của mình post trên mạng và like một post của bạn giống như cách trên Facebook, hãy đề xuất ý tưởng :

- a. Mô tả cấu trúc dữ liệu bạn nghĩ sẽ sử dụng phù hợp với các chức năng của mạng đã mô tả ở phần trên. (1 điểm)
  - Mục tiêu câu hỏi là kiểm tra khả năng vận dụng kiến thức của sinh viên đã học để giải quyết một bài toán cụ thể.
  - SV chỉ cần đề xuất ý tưởng sử dụng cấu trúc dữ liệu đã học để giải quyết và đề xuất ý tưởng các bước giải thuật để thực hiện yêu cầu của câu hỏi trên cơ sở tính phù hợp, hợp lý với cấu trúc dữ liệu sử dụng.
  - Câu hỏi KHÔNG yêu cầu về tính tối ưu của ý tưởng / giải pháp SV đề nghị sử dụng.

### Thang điểm đề nghị :

- SV liệt kê được các thông tin liên quan của 1 thành viên (user) như: ID, tên tài khoản, hình ảnh, password : 0.25 điểm
- SV trình bày được cài tổ chức lưu trữ : danh sách 1 thành viên (user); danh sách post của thành viên (user), danh sách mạng (có quan hệ giữa các thành viên) : 0.5 điểm
- SV viết được 1 cấu trúc dữ liệu cụ thể phục vụ chính xác cho nhu cầu lưu trữ của câu hỏi: 0.25 điểm - 0.5 điểm (tuỳ theo việc phân chia cấu trúc SV sử dụng phù hợp)

- SV mô tả bằng lời rất cụ thể việc sử dụng cấu trúc dữ liệu nào với mục đích lưu trữ cụ thể để làm gì cho yêu cầu của câu hỏi: 0.5 điểm.
- SV viết được code các cấu trúc hoàn chỉnh hoặc mô tả chính xác tất cả các cấu trúc sử dụng phục vụ chính xác, hợp lý cho yêu cầu của câu hỏi : full điểm

- Ví dụ một cách sử dụng cấu trúc mảng :

+ Cấu trúc lưu trữ thông tin người :

```
struct person
{
    int ID;
    char Name[];
}
```

+ Cấu trúc lưu trữ 1 post : 0.125 điểm

```
struct post
{
    char Content[]; // nội dung post
    char Date[]; // thời gian post
    int Count_like; // số like của post
    int ID_like[100]; // danh sách ID “bạn” đã like
}
```

+ Cấu trúc lưu trữ 1 thành viên :

```
struct member
{
    person PE; // thông tin thành viên
    post PO[1000]; // danh sách post
    int Number_post; // số lượng post
    person Friends[100]; // danh sách “bạn”
    int Number_friends; // số lượng “bạn”
}
```

+ Danh sách thành viên trong mạng :

```
member List[100];
```

- Ví dụ sử dụng cấu trúc danh sách liên kết :

+ SV có thể đề nghị sử dụng 100 danh sách liên kết ứng với 100 thành viên của mạng  
+ Mỗi danh sách có ID riêng tương ứng với 1 thành viên: các phần tử trong danh sách là các “bạn” của ID.  
+ Danh sách các post của mỗi thành viên có thể sử dụng tương tự cách sử dụng cấu trúc mảng

- Ví dụ sử dụng cấu trúc đồ thi :

+ SV có thể đề nghị sử dụng 1 đồ thi, trong đó cạnh là thể hiện mối quan hệ giữa 2 thành viên trong mạng có là “bạn” hay không (giá trị 0 hoặc 1 - ma trận kè), đỉnh đồ thị lưu thông tin của 1 thành viên  
+ Danh sách các post của mỗi thành viên có thể sử dụng tương tự cách sử dụng cấu trúc mảng

- Cách làm khác do SV đề nghị, Thầy, Cô thấy hợp lý ....
- b. Mô tả các giải thuật (các bước thực hiện giải thuật, không cài đặt) để hiện thực các chức năng của mạng đã mô tả ở phần trên và phù hợp với cấu trúc dữ liệu bạn đã đề nghị trong câu 4.a. (0.5 điểm)

### Thang điểm đề nghị :

- SV nhận diện / mô tả được các hàm tương ứng với chức năng tương ứng với mục đích cụ thể để làm gì : 0.25 điểm
- SV cụ thể hoá được input / output của các hàm, mô tả được cách làm của hàm : 0,25 điểm.

#### + Chức năng “kết bạn” :

Đầu vào : ID của thành viên, ID của “bạn” mới (cần kết bạn)

Đầu ra : Thông báo đã kết bạn thành công

Ý tưởng :

- Bước 1 : Kiểm tra ID “bạn” mới chưa có trong danh sách “bạn”
- Bước 2 : Kết bạn : cập nhật trong danh sách “bạn” của mình có thêm ID của “bạn” mới ( thông qua cấu trúc mảng, danh sách liên kết, đồ thị,...SV đã sử dụng)
- Bước 3 : Thông báo đã kết bạn thành công.

#### + Chức năng “like một post” :

Đầu vào : ID thành viên, Danh sách “bạn”

Đầu ra : Đã cập nhật like post của “bạn” thành công

Ý tưởng :

- Bước 1 : Hiển thị danh sách các post trong danh sách “bạn” của thành viên ( thông qua cấu trúc mảng, danh sách liên kết, đồ thị,... SV đã sử dụng)
- Bước 2 : Duyệt qua lần lượt các post cho đến hết danh sách post, nếu thích post nào đó chuyển qua bước 3, ngược lại qua bước 4
- Bước 3 : Like một post : cập nhật đã like, tăng số like cho post của “bạn” qua ID của “bạn” ( thông qua truy xuất cấu trúc mảng, danh sách liên kết, đồ thị,... SV đã sử dụng)
- Bước 4 : Quay lại bước 2

#### + Chức năng “duyệt các post” :

Đầu vào : ID thành viên, Danh sách “bạn”

Đầu ra : Đã duyệt các post của “bạn” thành công

Ý tưởng :

- Bước 1 : Hiển thị danh sách các post trong danh sách “bạn” của thành viên ( thông qua cấu trúc mảng, danh sách liên kết, đồ thị,... SV đã sử dụng)
- Bước 2 : Duyệt qua lần lượt các post cho đến hết danh sách post

HẾT

## Đề 1

### Câu 1:

1. Trình bày **các bước** của thuật toán sắp xếp **Insertion sort** cho thứ tự **tăng dần** 1 mảng số nguyên 1 chiều A gồm N phần tử.
2. Cho dãy số 10, 5, 30, 70, 40, 80, 90. Hãy trình bày từng bước trong quá trình sắp xếp dãy giảm dần của thuật toán **Insertion sort**

### Câu 2: Cho dãy số như sau: 30, 20, 24, 22, 40, 10, 23, 5, 35, 15, 48

1. Vẽ cây nhị phân tìm kiếm.
2. Cho biết kết quả duyệt RLN, NRL, RNL của cây NPTK ở trên.
3. Vẽ từng cây sau khi xóa lần lượt các nút sau: 24, 30, 40

### Câu 3: 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 có kích thước $M=10$

Cho 2 hàm băm như sau:  $h1(key) = key \% 10$  và  $h2(key) = (key \% 7)+1$

Hãy vẽ hình bảng băm khi thêm lần lượt các khóa trong K vào bảng băm. Trong trường hợp xảy ra đụng độ thì dùng phương pháp **băm kép** để giải quyết đụng độ với hàm băm  $h'(key)= ( h1(key)+ i*h2(key) ) \% 10$

### Câu 4: Tạo cây B-Tree bậc 3 từ dãy các khóa sau:

5, 30, 40, 70, 16, 82, 95, 100, 73, 54, 98, 37, 25, 62, 81, 150, 79, 181

Vẽ từng cây sau khi xóa lần lượt các node 150, 40, 100, 16

## Đề 2

Câu 1:

1. Trình bày **code** của thuật toán sắp xếp **Quick sort** cho thứ tự **tăng dần** 1 mảng số nguyên 1 chiều A gồm N phần tử.
2. Cho dãy số 2, 5, 4, 7, 3, 9, 1, 8. Hãy trình bày từng bước trong quá trình sắp xếp dãy giảm dần của thuật toán **Quick sort**

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

1. Vẽ cây nhị phân tìm kiếm.
2. Vẽ từng cây sau khi xóa lần lượt các nút sau: D, E, F, H.
3. Viết hàm tính tổng giá trị các node có 1 con.

Câu 3: Cho K là một tập các giá trị khóa là số nguyên như sau:  $K=\{Y, D, d, p, O, X, c\}$  và bảng băm có kích thước  $M=11$

Hãy vẽ hình bảng băm khi thêm lần lượt các khóa trong K vào bảng băm. Trong trường hợp xảy ra đụng độ thì dùng phương pháp **dò tuyển tính** để giải quyết đụng độ.

Câu 4: Tạo cây B-Tree bậc 5 từ dãy các khóa sau:

5, 30, 40, 70, 16, 82, 95, 100, 73, 54, 98, 37, 25, 62, 81, 150, 79, 181

Vẽ từng cây sau khi xóa lần lượt các node 30, 54, 70, 25

Môn thi: CTDL & GT

Mã lớp: Các lớp đại trà, chất lượng cao

Thời gian làm bài: 90 phút

(Sinh viên không được sử dụng tài liệu)

### Câu 1: 1.5 điểm

- Hãy cho biết độ phức tạp của thuật toán Selection sort theo định nghĩa Big-O (O lớn) (0.25 điểm)
- Viết hàm sắp xếp mảng 1 chiều giảm dần với thuật toán Selection sort (0.5 điểm)
- Chạy từng bước thuật toán đã viết ở trên với dãy số sau: 5, 8, 9, 10, 3, 6 (0.75 điểm)

### Câu 2: 3.5 điểm

Cho dãy ký tự như sau: F, D, B, A, C, E, H, G, 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. (1 điểm)
- Cho biết kết quả duyệt cây theo RNL, NRL. (0.5 điểm)
- Huỷ lần lượt từng nút D, E, F, H trên cây, mỗi lần huỷ 1 nút vẽ lại cây nối tiếp theo như thứ tự huỷ. (1 điểm)
- Viết hàm đếm số lượng nút có một nút con trên cây, nếu cây rỗng thì in ra giá trị -1. (1 điểm)

### Câu 3: 2 điểm

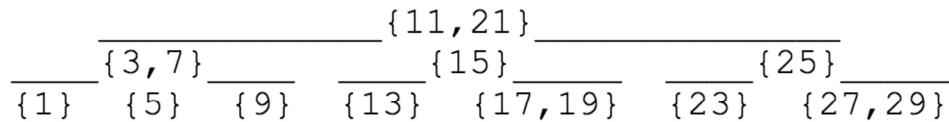
Cho biết cây B-Tree bậc 3 là một cây thỏa mãn các tính chất sau:

- Tất cả node lá nằm trên cùng một mức
- Tất cả các node, trừ node gốc và node lá, có **\*tối thiểu\*** 2 node con.
- Tất cả các node có **\*tối đa\*** 3 con
- Tất cả các node, trừ node gốc, có từ 1 cho đến 2 khóa (keys)
- Một node không phải lá và có n khóa thì phải có n+1 node con.

Hãy thực hiện các yêu cầu sau:

- 3.1 Cho dãy số: 27, 19, 23, 9, 1, 3, 11, 21, 5, 13, 17, 15, 29, 25. Hỏi khi lần lượt thêm các số trong dãy theo thứ tự từ trái qua phải vào một cây B-Tree bậc 3 rỗng thì:
- Các khóa nào khi thêm vào sẽ làm phát sinh thao tác split node? (0.5 điểm)
  - Vẽ cây B-Tree trước và sau khi thêm các khóa trên. (1 điểm)

- 3.2 Cho cây B-Tree bậc 3 như hình sau:



Hãy lần lượt tiến hành xóa các khóa sau khỏi cây B-Tree: 11, 21, 13 và vẽ cây B-Tree trước sau khi xóa mỗi khóa trên. (0.5 điểm)

Lưu ý khi xoá:

- Khi khóa cần xóa (gọi là x) không nằm ở node lá, chọn khóa thế mạng là khóa có giá trị lớn nhất mà nhỏ hơn x.
- Thao tác underflow sẽ được thực hiện khi hai node liền kề có tổng số khóa  $\geq 2$ . Khi có một node không còn đáp ứng đủ số lượng khóa tối thiểu, luôn ưu tiên thực hiện underflow thay cho catenation vì thao tác này không làm thay đổi số khóa của node cha.
- Khi có 02 lựa chọn node liền kề để thực hiện catenation, ưu tiên chọn catenate giữa node bị thiếu khóa với node liền trước

#### Câu 4: 2 điểm

Cho một bảng băm theo phương pháp thăm dò bậc 2 với hàm băm  $h(\text{key})$  và hàm băm lại (hay hàm thăm dò)  $prob(\text{key}, i)$  như sau:

$$h(\text{key}) = (\text{key \% } M) \quad prob(\text{key}, i) = (h(\text{key}) + i*i) \% M$$

Trong đó:

- key là giá trị khóa.
- i là một số nguyên cho biết lần thăm dò thứ i.
- M là kích thước bảng băm.

Cho  $M = 7$  và trên bảng băm đã chứa các mục dữ liệu như bên dưới. Biết EMP và DEL lần lượt là ký hiệu để đánh dấu vị trí còn trống hoặc đã bị xóa trong bảng băm.

Key	
0	EMP
1	EMP
2	2
3	EMP
4	4
5	EMP
6	EMP

- a. Trình bày từng bước việc thêm các khóa **Key** trong danh sách bên dưới vào bảng băm theo đúng thứ tự trong danh sách. (1 điểm)

STT	Key
1	6
2	16
3	10

- b. Trình bày từng bước việc xóa giá trị **Key**=16 trong bảng băm khi hoàn thành yêu cầu ở câu a. (0.5 điểm)
- c. Trình bày từng bước việc tìm giá trị **Key**=10 trong bảng băm khi hoàn thành yêu cầu ở câu b. (0.5 điểm)

#### Câu 5: 1 điểm

Cho bài toán “Tô màu bản đồ” được đặt ra như sau: Có một bản đồ các quốc gia trên thế giới, ta muốn tô màu các quốc gia này sao cho hai nước có cùng ranh giới được tô khác màu nhau. Yêu cầu tìm cách tô sao cho số màu sử dụng là ít nhất. Bài toán có thể được mô hình hóa thành một bài toán trên đồ thị, khi đó mỗi nước trên bản đồ là

một đỉnh của đồ thị, hai nước láng giềng tương ứng với hai đỉnh kề nhau được nối với nhau bằng một cạnh, bài toán trở thành: tô màu các đỉnh của đồ thị sao cho mỗi đỉnh chỉ được tô một màu, hai đỉnh kề nhau có màu khác nhau và số màu sử dụng là ít nhất.

Giả sử cho thông tin đầu vào của bài toán được nhập vào chương trình như sau:

Ví dụ Input	Giải thích
15	- Dòng đầu tiên chứa một số <i>e</i> là số cạnh của đồ thị
Viet_Nam Lao	- <i>e</i> dòng tiếp theo, mỗi dòng chứa 02 chuỗi <i>u</i> và <i>i</i> , thể hiện thông tin có một cạnh nối từ đỉnh <i>u</i> sang đỉnh <i>i</i> trong đồ thị
Viet_Nam Trung_Quoc	
Thai_Lan Lao	
...	
Campuchia Thai_Lan	Lưu ý: không biết trước số đỉnh và danh sách các đỉnh

Hãy thực hiện các yêu cầu sau:

- Xây dựng cấu trúc dữ liệu thích hợp để biểu diễn đồ thị nhằm lưu trữ các thông tin cần thiết trên bản đồ. (0.5 điểm)
- Viết hàm nhập đồ thị (bằng cách nhập số cạnh và danh sách các cạnh như ví dụ ở trên) và lưu trữ thông tin của đồ thị vào cấu trúc dữ liệu đã đề xuất ở câu a. (0.5 điểm)

**Hết**

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

**Thang điểm gợi ý :**

Câu	Nội dung	Điểm
1.a	Cho biết độ phức tạp của thuật toán Insertion sort	0.25 điểm
1.b	Viết đúng hàm sắp xếp mảng 1 chiều N phần tử giảm dần theo thuật toán insertion sort	0.75 điểm
1.c	Mô tả dãy số thay đổi qua từng bước thuật toán	1 điểm
2.a	Vẽ cây nhị phân tìm kiếm đúng	1 điểm
2.b	Duyệt cây theo RNL, NRL	1 điểm
2.c	Hủy lần lượt từng nút L, T, E, R trên cây	1 điểm
3.1.a	Xác định các khóa khi thêm vào cây sẽ làm phát sinh thao tác tách node	0.5 điểm
3.1.b	Vẽ cây B-Tree trước và sau khi thêm các khóa ở 3.1.a	1 điểm
3.2	Xóa các khóa khỏi cây: 13, 24, 19 và vẽ cây B-Tree trước và sau khi xóa mỗi khóa	0.5 điểm
4.a	Trình bày từng bước việc tìm mã quản lý 23 trong bảng băm	0.5 điểm
4.b	Trình bày từng bước việc thêm các mã quản lý 11, 20, 27 vào bảng băm theo đúng thứ tự	1.5 điểm
5.a	Định nghĩa CTDL phù hợp	0.5 điểm
5.b	Viết hàm nhập theo Input và lưu trữ thông tin của đồ thị vào cấu trúc dữ liệu ở 5.a	0.5 điểm

Tổng điểm : 10

**Câu 1:**

- a. Hãy cho biết độ phức tạp của thuật toán Insertion sort (chèn trực tiếp) theo định nghĩa Big-O (O lớn) (0.25 điểm)

**Đáp án tham khảo:**

Độ phức tạp thuật toán trong 3 trường hợp: tốt nhất, xấu nhất, TB là  $O(n^2)$

- b. Viết hàm sắp xếp mảng 1 chiều gồm N phần tử giảm dần với thuật toán Insertion sort (0.75 điểm)

**Đáp án tham khảo:**

<pre>void InsertionSort(int a[], int n) {     for(int i=1;i&lt;=n-1;i++)     {         int x = a[i];         for(int j=i-1; j&gt;=0&amp;&amp;a[j]&lt;x;j--)             a[j+1]=a[j];         a[j+1] = x;     } }</pre>	<p>// đúng vòng lặp for/while thứ nhất : 0.25đ</p> <p>// đúng vòng lặp for/while thứ hai: 0.25đ</p> <p>//đúng phép gán 0.25đ</p>
--	--

- c. Hãy cho biết dãy số sẽ thay đổi qua từng bước như thế nào khi áp dụng thuật toán ở câu 1b, biết rằng dãy số cho như sau: 3, 8, 4, 5, 9, 1, 2, 6 (1 điểm)

**Đáp án tham khảo:**

Bước	Mảng cần sắp xếp	Ghi chú
1	8 3 4 5 9 1 2 6	Chèn 8 đúng – 0.25 điểm
2	8 4 3 5 9 1 2 6	Chèn 4 đúng – 0.125 điểm
3	8 5 4 3 9 1 2 6	Chèn 5 đúng – 0.125 điểm
4	9 8 5 4 3 1 2 6	Chèn 9 đúng – 0.125 điểm
5	9 8 5 4 3 1 2 6	Chèn 1 đúng – 0.125 điểm
6	9 8 5 4 3 2 1 6	Chèn 2 đúng – 0.125 điểm
7	9 8 6 5 4 3 2 1	Chèn 6 đúng – 0.125 điểm

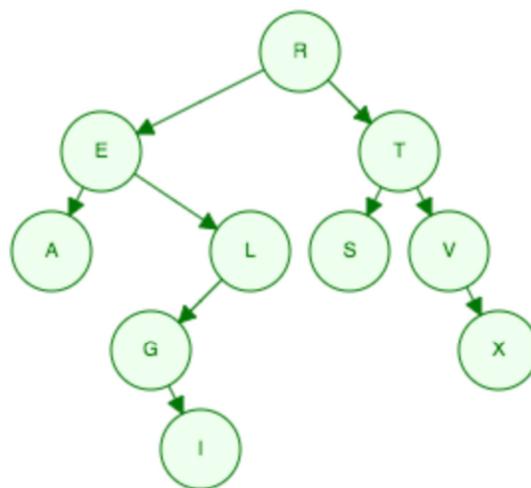
## Câu 2:

Cho dãy ký tự như sau: R, E, T, A, V, X, L, G, S, I

Hãy thực hiện các yêu cầu sau:

- a. Vẽ cây nhị phân 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 (1 điểm)

**Đáp án tham khảo:**



- b. Cho biết kết quả duyệt cây theo RNL, NRL (1 điểm)

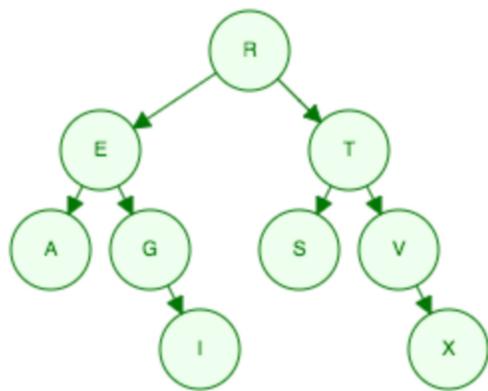
**Đáp án tham khảo:**

- Duyệt cây RNL – 0.5 điểm: X,V,T,S,R,L,I,G,E,A
- Duyệt cây NRL – 0.5 điểm: R,T,V,X,S,E,L,G,I,A

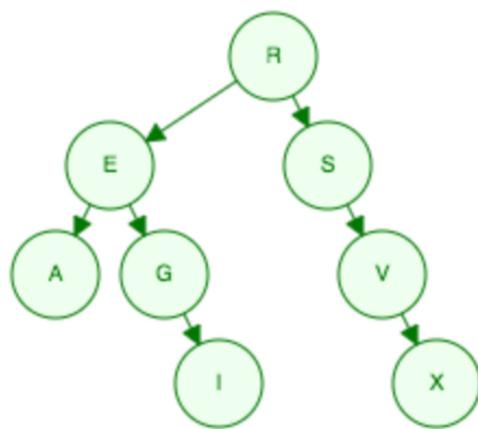
- c. Huỷ lần lượt từng nút L, T, E, R trên cây, mỗi lần huỷ 1 nút vẽ lại cây nối tiếp theo như thứ tự huỷ (1 điểm)

**Đáp án tham khảo:**

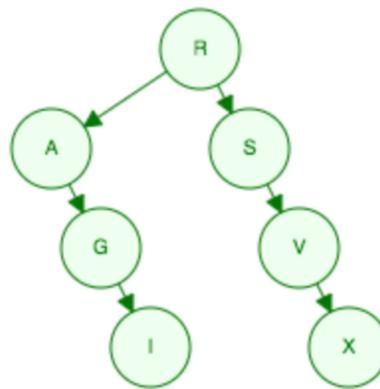
- Xoá L – 0.25 điểm



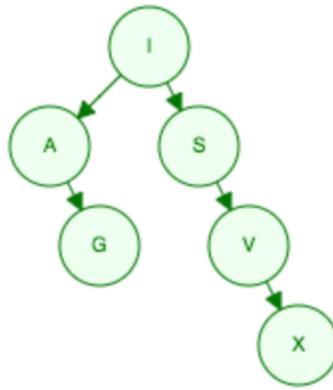
- Xoá T – 0.25 điểm



- Xoá E – 0.25 điểm



- Xoá A – 0.25 điểm



### Câu 3:

Cho biết cây B-Tree bậc 3 là một cây thỏa mãn các tính chất sau:

- Tất cả node lá nằm trên cùng một mức
- Tất cả các node, trừ node gốc và node lá, có **\*tối thiểu\*** 2 node con.
- Tất cả các node có **\*tối đa\*** 3 con
- Tất cả các node, trừ node gốc, có từ 1 cho đến 2 khóa (keys)
- Một node không phải lá và có n khóa thì phải có n+1 node con.

Hãy thực hiện các yêu cầu sau:

3.1 Cho dãy số: 12, 17, 20, 23, 15, 11, 24, 13, 19, 22, 18, 21, 16. Hỏi khi lần lượt thêm các số trong dãy theo thứ tự từ trái qua phải vào một cây B-Tree bậc 3 rỗng thì:

- a. Các khóa nào khi thêm vào cây sẽ làm phát sinh thao tác tách (split) node? (0.5 điểm)

Đáp án tham khảo:

Các khóa làm phát sinh thao tác split node là: 20, 11, 24, 22, 16 (0.5 điểm)

*sinh viên có thể ghi không theo thứ tự trên nhưng đảm bảo chính xác giá trị của 05 khóa trên thì vẫn được 0.5 điểm*

- b. Vẽ cây B-Tree trước và sau khi thêm các khóa ở câu a (1 điểm)

Đáp án tham khảo:

*Sinh viên không cần vẽ chi tiết từng bước thực hiện thao tác split, chỉ cần vẽ trạng thái cây trước khi tiến hành thêm khóa và cây sau khi thêm khóa vào hoàn tất. Tổng cộng có 10 hình cần vẽ. Ở câu này sinh viên bắt buộc phải làm theo thứ tự các khóa sẽ được thêm. Khi vẽ sai ở bước nào thì các bước sau sẽ không được điểm.*

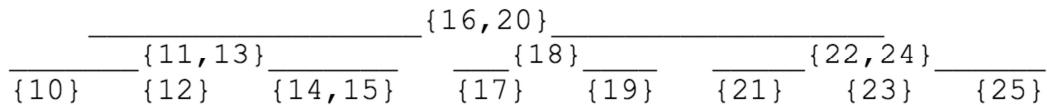
Khóa	trước	Sau
20	{12, 17}	$\overline{\{17\}}$ $\overline{\{12\}} \quad \overline{\{20\}}$
11	$\overline{\{17\}}$ $\overline{\{12, 15\}} \quad \overline{\{20, 23\}}$	$\overline{\{12, 17\}}$ $\overline{\{11\}} \quad \overline{\{15\}} \quad \overline{\{20, 23\}}$
24	$\overline{\{12, 17\}}$ $\overline{\{11\}} \quad \overline{\{15\}} \quad \overline{\{20, 23\}}$	$\overline{\{12\}} \quad \overline{\{17\}}$ $\overline{\{11\}} \quad \overline{\{15\}} \quad \overline{\{20\}} \quad \overline{\{23\}} \quad \overline{\{24\}}$
22	$\overline{\{12\}}$ $\overline{\{17\}}$ $\overline{\{11\}}$ $\overline{\{13, 15\}}$ $\overline{\{23\}}$ $\overline{\{19, 20\}}$ $\overline{\{24\}}$	$\overline{\{12\}}$ $\overline{\{17\}}$ $\overline{\{11\}}$ $\overline{\{13, 15\}}$ $\overline{\{19\}}$ $\overline{\{22\}}$ $\overline{\{20, 23\}}$ $\overline{\{24\}}$
16	$\overline{\{12\}}$ $\overline{\{17\}}$ $\overline{\{11\}}$ $\overline{\{13, 15\}}$ $\overline{\{20, 23\}}$ $\overline{\{18, 19\}}$ $\overline{\{21, 22\}}$ $\overline{\{24\}}$	$\overline{\{12, 15\}}$ $\overline{\{17\}}$ $\overline{\{11\}}$ $\overline{\{13\}}$ $\overline{\{16\}}$ $\overline{\{18, 19\}}$ $\overline{\{21, 22\}}$ $\overline{\{24\}}$

**Thang điểm:**

Vẽ đúng TOÀN BỘ từ trước khi thêm khóa 20 đến sau khi thêm khóa 24: 0.5 điểm

Vẽ đúng hình cả trước và sau khi thêm khóa 22: 0.25 điểm, đúng hình cả trước và sau khi thêm khóa 16: 0.25 điểm

3.2 Cho cây B-Tree bậc 3 như hình sau:



Hãy lần lượt tiến hành xóa các khóa sau khỏi cây: 13, 24, 19 và vẽ cây B-Tree trước và sau khi xóa mỗi khóa trên (0.5 điểm)

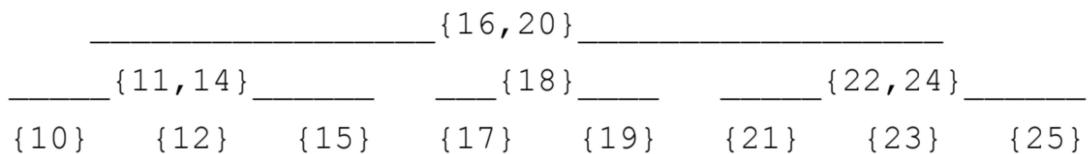
**Lưu ý khi xoá:**

- Khi khóa cần xóa (gọi là x) không nằm ở node lá, chọn khóa thê mạng là khóa có giá trị lớn nhất mà nhỏ hơn x.
- Thao tác nhường khoá (underflow) sẽ được thực hiện khi hai node liền kề có tổng số khóa  $\geq 2$ . Khi có một node không còn đáp ứng đủ số lượng khóa tối thiểu, ưu tiên thực hiện underflow thay cho catenation (hợp) vì thao tác này không làm thay đổi số khóa của node cha.
- Khi có 02 lựa chọn node liền kề để thực hiện catenation, ưu tiên chọn catenation giữa node bị thiếu khóa với node liền trước.

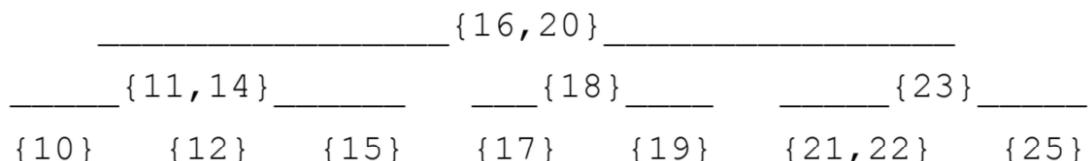
**Đáp án tham khảo:**

Câu này bắt buộc phải làm theo thứ tự xóa, nếu xóa sai ở khóa nào thì các bước tiếp theo chắc chắn xem như sai. Đề thi đã quy định rõ các quy ước cần tuân theo khi xóa nên chỉ có duy nhất 01 đáp án đúng có thể được chấp nhận.

- Xóa 13: swap 13 và 12 sau đó xóa. Ưu tiên thực hiện underflow giữa node {14, 15} và node vừa bị thiếu khóa thay vì catenate nó với node {10}. Xóa đúng được 0.125 điểm



- Xóa 24 - Swap 24 với 23 và thực hiện catenate, ưu tiên catenate với {21} thay vì {25}. Xóa đúng được 0.125 điểm



- Xóa 19 trực tiếp tại lá sau đó xóa và catenate node bị xóa với {17} làm {18} bị thiếu khóa và lan truyền xử lý này lên khiến node {11, 14} phải underflow cho node bị thiếu khóa. Xóa đúng được 0.25 điểm

		{14, 20}	
{11}	{16}	{23}	
{10}	{12}	{15}	{17, 18}
			{21, 22}
			{25}

#### Câu 4:

Để việc tìm kiếm thông tin mặt hàng được nhanh chóng, người ta dùng một bảng băm theo phương pháp thăm dò, làm việc trên mã quản lý của mặt hàng. Mã quản lý này là một con số nguyên. Bảng băm có:

- Hàm băm:  $h(\text{key}) = (\text{key} \% M)$
- Hàm băm lại (hàm thăm dò):  $\text{prob}(\text{key}, i) = (h(\text{key}) + i*i + i) \% M$

Trong đó:

- key là giá trị khóa.
- i là một số nguyên cho biết lần băm lại (thăm dò) thứ i.
- M là kích thước bảng băm.

Giả sử  $M = 7$ , cho trường hợp  $T$  của bảng băm đã chứa dữ liệu như bên dưới. Biết “-” là ký hiệu vị trí trống trong bảng băm.

Bảng băm $T$	
0	-
1	-
2	16
3	-
4	-
5	12
6	13

- a. Trình bày từng bước việc tìm mã quản lý 23 trong bảng băm  $T$ . (0.5 điểm)

Đáp án tham khảo:

- Tính hàm băm với khóa 23, được vị trí 2 đang chứa giá trị 16 khác 23  $\rightarrow$  băm lại (**0.25 điểm cho việc so sánh khóa và băm lại**)
- Băm lại lần 1, được vị trí 4 đang chứa giá trị “-”  $\rightarrow$  không tìm thấy khóa 23 (**0.25 điểm cho việc xác định kết quả không tìm thấy khi gặp vị trí trống**)

- b. Trình bày từng bước việc thêm các mã quản lý sau vào bảng băm  $T$  theo đúng thứ tự liệt kê là 11, 20, 27 (1.5 điểm).

Đáp án tham khảo:

- Thêm khóa 11:
- Tính hàm băm với khóa 11, được vị trí 4 đang chứa giá trị “-”  $\rightarrow$  Thêm 11 vào vị trí 4 (**0.25 điểm cho việc tính đúng vị trí 4 và 0.25 điểm cho thao tác thêm khi gặp vị trí trống**)

- Bảng băm  $T$

0	-
1	-
2	16
3	-

4	11
5	12
6	13

- Thêm khóa 20:
- Tính hàm băm với khóa 20, được vị trí 6, đúng độ → băm lại (*0.25 điểm cho việc xác định đúng độ và băm lại*)
- Băm lại lần 1, được vị trí 1 đang chứa giá trị “-“, → thêm 20 vào vị trí 1 (*0.25 điểm cho việc tính đúng vị trí và thêm khi gặp vị trí trống*)

- Bảng băm  $T$

0	-
1	20
2	16
3	-
4	4
5	12
6	13

- Thêm khóa 27:
- Tính hàm băm với khóa 27, được vị trí 6, đúng độ → băm lại
- Băm lại lần 1, được vị trí 1, đúng độ → băm lại
- Băm lại lần 2, được vị trí 5, đúng độ → băm lại
- Băm lại lần 3, được vị trí 4, đúng độ → băm lại
- Băm lại lần 4, được vị trí 5, đúng độ → băm lại
- Băm lại lần 5, được vị trí 1, đúng độ → băm lại
- Băm lại lần 6, được vị trí 6, đúng độ → băm lại (*0.25 điểm cho việc thực hiện băm lại 6 lần*)
- Đã băm lại  $M-1 = 6$  lần → kết luận không thêm được / không tìm được vị trí thêm / không tìm được chỗ trống (*0.25 điểm cho việc không thêm được khi băm lại M-1 lần*)

### Câu 5:

Trong các ứng dụng thực tế, chẳng hạn trong mạng lưới giao thông đường bộ, đường thủy hoặc đường hàng không, người ta không chỉ quan tâm đến việc tìm đường đi giữa hai địa điểm mà còn phải lựa chọn một hành trình tiết kiệm nhất (theo tiêu chuẩn không gian, thời gian hay chi phí). Vấn đề này có thể được mô hình hóa thành một bài toán trên đồ thị, trong đó mỗi địa điểm được biểu diễn bởi một đỉnh, cạnh nối hai đỉnh biểu diễn cho “đường đi trực tiếp” giữa hai địa điểm (tức không đi qua địa điểm trung gian) và trọng số của cạnh là khoảng cách giữa hai địa điểm.

Bài toán có thể phát biểu dưới dạng tổng quát như sau: Cho một đơn đồ thị có hướng và có trọng số dương  $G=(V,E)$ , trong đó  $V$  là tập đỉnh,  $E$  là tập cạnh (cung) và các cạnh đều có trọng số, hãy tìm một đường đi (không có đỉnh lặp lại) ngắn nhất từ đỉnh xuất phát  $S$  thuộc  $V$  đến đỉnh đích  $F$  thuộc  $V$ .

Giả sử thông tin đầu vào của bài toán (Input) được nhập vào chương trình như sau:

Input	Giải thích
7	- Dòng đầu tiên chứa một số nguyên dương $e$ cho biết số cạnh của đồ thị
A B 1	- Với $e$ dòng tiếp theo, mỗi dòng chứa hai chuỗi $u, i$ và một số nguyên
B E 3	dương $x$ , thể hiện thông tin có một cạnh nối từ đỉnh $u$ sang đỉnh $i$ trong
E D 3	đồ thị với độ dài (trọng số) là $x$

C B 4 A D 7 E C 2 C D 1 A E	<p>- Dòng cuối cùng chứa hai chuỗi s và f, đây là đỉnh bắt đầu và đỉnh kết thúc của đường đi cần tìm</p> <p><b>Lưu ý: không biết trước số đỉnh và danh sách các đỉnh.</b></p>
---	---

Hãy thực hiện các yêu cầu sau:

- a. Xây dựng các cấu trúc dữ liệu phù hợp nhất có thể để biểu diễn đồ thị trên máy tính theo input đã cho. (0.5 điểm)  
Cấu trúc được xem là tốt nếu đạt được các tiêu chuẩn sau: Tiết kiệm tài nguyên; Hỗ trợ một số thao tác cơ bản như “Kiểm tra hai đỉnh có kề nhau không”, “Tìm danh sách các đỉnh kề với một đỉnh cho trước” với ràng buộc là không phải duyệt qua danh sách tất cả các cạnh của đồ thị.
- b. Viết hàm nhập theo Input ở đầu bài và lưu trữ thông tin của đồ thị vào cấu trúc dữ liệu đã đề xuất ở câu a. (0.5 điểm)

**\*\*\* KHÔNG YÊU CẦU tìm cách giải cho bài toán này. Sinh viên ĐUQC PHÉP sử dụng Standard Template Library-STL với những cấu trúc dữ liệu (vector, stack, queue, list, map, set, pair, ...) cũng như giải thuật được xây dựng sẵn.**

#### **Đáp án tham khảo:**

##### **I. Một số lưu ý:**

###### **1. Đây là câu hỏi có tính thử thách cao để phân loại sinh viên giỏi.**

Nếu như “Bài toán tìm đường đi” chưa được giới thiệu trên lớp/hoặc không có trong slide bài giảng chung thì cũng không vấn đề gì, vì đề bài có mô tả bài toán rõ ràng, không yêu cầu SV tìm cách giải, đề thi kiểm tra về “cách biểu diễn đồ thị trên máy tính, ưu và nhược điểm của mỗi phương pháp”.

###### **2. Việc chọn cấu trúc dữ liệu nào để biểu diễn đồ thị phụ thuộc vào từng tình huống cụ thể và có tác động rất lớn đến hiệu quả của thuật toán.**

Để tránh trường hợp SV “sáng tạo” những cách biểu diễn không phù hợp như lưu toàn bộ input thành chuỗi, hoặc tạo nhiều biến/mảng không cần thiết, đề bài có ghi rõ tiêu chuẩn đánh giá. Nếu không có tiêu chuẩn rõ ràng thì GV rất khó chấm điểm.

###### **3. Khi trình bày CTDL cần làm rõ: khai báo kiểu dữ liệu mô tả cho các đối tượng thông tin trong bài toán; Cho biết ý nghĩa, vai trò của từng biến, mảng, cấu trúc, thuộc tính của cấu trúc, lớp và thuộc tính của lớp.... được sử dụng → Nếu SV không giải thích gì, người chấm đọc không hiểu và dễ gây nhầm lẫn thì có thể TRỪ ĐIỂM**

##### **II. CTDL phù hợp cho tình huống đã đặt ra**

Input không cho biết trước số đỉnh và danh sách các đỉnh và trên thực tế đồ thị cho bài toán này thường sẽ thưa (sparse). Cách tổ chức đồ thị hợp lý nhất là danh sách kề. Danh sách kề có rất nhiều cách cài đặt. Nếu SV dùng danh sách kề, trong đó mô tả CTDL rõ ràng, viết hàm nhập chấp nhận được thì trọn 1 điểm

##### **III. Các cách biểu diễn khác**

Một số cách biểu diễn khác dù không đáp ứng đầy đủ các tiêu chuẩn trong đề bài nhưng vẫn có thể chấp nhận và cho điểm, **nhưng chỉ được tối đa 0.5/1 điểm**

###### **1. Dùng ma trận kề, ma trận trọng số**

Cách biểu diễn này cần biết trước số đỉnh (để khai báo kích thước ma trận) và danh sách các đỉnh (để có thể ánh xạ từ mỗi đỉnh sang index dòng/cột tương ứng trong ma

trận). Nếu khai báo kích thước ma trận lớn gây lãng phí bộ nhớ (đặc biệt khi đồ thị thưa)

## 2. Dùng danh sách cạnh

Nhược điểm là để xác định những đỉnh nào của đồ thị là kè với một đỉnh cho trước chúng ta phải duyệt qua danh sách tất cả các cạnh của đồ thị

Bảng thang điểm gợi ý chi tiết cho 2 câu 5.a và 5.b

Cách thực hiện sử dụng :	Mô tả	Thang điểm Thiết kế CTDL (5.a)	Thang điểm Hàm nhập theo input (5.b)
Danh sách kè	<p>Ứng với mỗi đỉnh i, ta cần lưu trữ một tập hợp (danh sách) gồm các đỉnh kè với đỉnh i</p> <p><b>Có nhiều cách cài đặt</b></p> <p><b>Ví dụ: Cài đặt Danh sách kè dùng map trong STL</b></p> <pre>map&lt;string, set&lt;pair&lt;string,int&gt;&gt; adj_list</pre> <p>Có thể dùng vector, list, tree thay cho set cũng được.</p> <p>A → {(B, 1), (D, 7)}</p> <p>B → {(E, 3)}</p> <p><b>Hoặc</b></p> <pre>map&lt;string, map&lt;string,int&gt;&gt; adj_list</pre> <p>A--&gt; [B--&gt;10] [O--&gt;45] [C--&gt;12]</p> <p>B--&gt; [C--&gt;4] [L--&gt;8]</p> <p><b>Ví dụ: Cài đặt Danh sách kè dùng các danh sách liên kết</b></p> <p>vector&lt;list&lt;node&gt;&gt; adj_list và 1 cấu trúc để tra cứu từ chuỗi sang index. Struct node chứa thông tin tên đỉnh và trọng số.</p>	0.5 điểm	0.5 điểm
Ma trận trọng số	<ul style="list-style-type: none"> <li>- Giá trị của mỗi ô trong ma trận nếu khác 0 thì cho biết trọng số của cạnh nối giữa 2 đỉnh tương ứng</li> <li>- Do đỉnh có tên là chuỗi nên cần có cách ánh xạ từ chuỗi sang số nguyên cho biết index dòng/cột tương ứng trong ma trận, <b>nếu SV không xử lý được thì không có điểm</b></li> </ul> <p>Ví dụ: cách ánh xạ từ chuỗi sang index</p> <ol style="list-style-type: none"> <li>1. map&lt;string, int&gt; name_to_index;</li> </ol>		

	<p>hoặc</p> <p>2. dùng mảng lưu các đỉnh (khi muốn biết đỉnh tương ứng với index nào thì phải search)</p>		
	<p>Nếu SV có cách xử lý đếm được số đỉnh của đồ thị và khai báo kích thước ma trận phù hợp thì có thể đạt tối đa 1 điểm</p> <p>Ví dụ:</p> <p><b>Cách 1</b></p> <pre>vector&lt;string&gt; nodes; vector&lt;pair&lt;string, string&gt;&gt; edges;</pre> <p>Khi nhập cạnh thì kiểm tra 2 đỉnh có trong nodes chưa, nếu chưa thì thêm đỉnh mới vào và thêm cạnh vào edges</p> <pre>int nums=nodes.size() vector&lt;vector&lt;int&gt;&gt; matrix (num, vector&lt;int&gt;(num,0))</pre> <p><b>Cách 2:</b> cấp phát động và thêm dòng, thêm cột khi có nhu cầu</p>	<b>0.5 điểm</b>	<b>0.5 điểm</b>
	<p>Nếu chỉ define MAX 2000 thì kích thước không phù hợp, chỉ đạt tối đa 0.5</p> <pre>vector &lt;vector&lt;int&gt; &gt; matrix (Max, vector&lt;int&gt; (Max, 0)); hoặc int **matrix; hoặc int matrix[MAX][MAX]</pre>	<b>0.25 điểm</b>	<b>0.25 điểm</b>
Danh sách cạnh	<p>Lưu các bộ 3 (x,y,w) tương ứng đỉnh đầu, đỉnh cuối và trọng số của các cạnh.</p> <p>Ví dụ:</p> <ol style="list-style-type: none"> <li>map&lt;pair&lt;string, string&gt;, int&gt; (A,B)--&gt;1 (A,C)--&gt;2 (A,D)--&gt;4</li> </ol> <p>Hoặc</p> <ol style="list-style-type: none"> <li>Dùng 1 ma trận có m dòng 2 cột lưu m cạnh và thêm 1 mảng 1 chiều khác lưu m trọng số</li> </ol> <p>Hoặc</p> <ol style="list-style-type: none"> <li>Dùng mảng/danh sách... kiểu node, mỗi node có 3 thuộc tính (x,y,w) ...</li> </ol>	<b>0.25 điểm</b>	<b>0.25 điểm</b>

Hết

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**ĐỀ THI CUỐI KỲ MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**  
**Thời gian: 90 phút**  
**(Không sử dụng tài liệu)**

**Câu 1 (3 điểm) :** Cho dãy số ban đầu như sau : 17 72 99 32 58 70 44 12 23

Hãy thực hiện các yêu cầu sau:

- Hãy trình bày các bước thực hiện **thuật toán chọn trực tiếp** (1.5 đ)
- Vẽ hình từng bước thực hiện của thuật toán trên để sắp xếp dãy số theo thứ tự **giảm dần** (không cần lập trình) (1.5 đ)

**Gợi ý đáp án:**

- Sinh viên trình bày lý thuyết đúng thuật toán (1,5 điểm)

Bước 1:  $i = 0$ ;

Bước 2: Tìm phần tử  $a[min]$  nhỏ nhất trong dãy hiện hành từ  $a[i]$  đến  $a[N-1]$

Bước 3 : Hoán vị  $a[min]$  và  $a[i]$

Bước 4 : Nếu  $i < N-2$  thì  $i = i+1$ ; Lặp lại Bước 2

Ngược lại: Dừng. //N-1 phần tử đã nằm đúng vị trí.

- Viết đúng các bước đưa từng phần tử nhỏ/lớn nhất lên trước (1,5 điểm). Nếu SV làm sai phương pháp thì không tính điểm.

Nếu SV lỡ nhầm sai 1-2 trường hợp thì chỉ cho 1.0 điểm

---

**Câu 2 (4 điểm) :** Cho dãy các ký tự như sau : A B C D E F W Z U T K

Hãy thực hiện các yêu cầu sau:

- Hãy vẽ cây nhị phân tìm kiếm từ dãy ký tự trên (1 đ)
- Bổ xung lần lượt các ký tự sau vào cây N, G, H , M , L để hình thành cây nhị phân tìm kiếm mới, vẽ hình cây khi thêm từng ký tự vào cây (1 đ)
- Trình bày dãy ký tự kết quả khi duyệt cây theo thứ tự NRL, LRN (1 đ)
- Vẽ hình cây khi xóa lần lượt các ký tự W, E, H, C (1 đ)

**Lưu ý :** trong quá trình bổ xung hay xóa một nút trên cây, nếu có xảy ra mất cân bằng thì cho biết trường hợp mất cân bằng là loại gì và tiến hành cân bằng lại cây.

**Câu 3 (2 điểm) :** Cho một danh sách liên kết đôi đã lưu thông tin về sản phẩm trong một công ty, bao gồm:

- 1.Mã sản phẩm (kiểu số nguyên)
- 2.Tên sản phẩm (kiểu chuỗi)
- 3.Chủng loại (bằng Giấy, bằng Kim loại, bằng Nhựa)
- 4.Năm sản xuất (kiểu số nguyên)
- 5.Số năm bảo hành (kiểu số nguyên)

Hai con trỏ Head, Tail đang trỏ đến phần tử đầu tiên và cuối cùng trong danh sách trên.

Hãy thực hiện các yêu cầu sau :

- a. Viết hàm sắp xếp các sản phẩm theo mã sản phẩm giảm dần (1 đ)
- b. Viết hàm xóa các sản phẩm đã hết hạn bảo hành ra khỏi danh sách khi thỏa điều kiện : Năm sản xuất + Số năm bảo hành > Năm hiện tại (1 đ)

**Gợi ý đáp án:**

- a.
  - b.
- 

**Câu 4 (1 điểm) :** Trình bày ngắn gọn ý tưởng về bảng băm. Cho biết bảng băm tối ưu hơn các cấu trúc dữ liệu đã học nào và cho một ví dụ minh họa.

**Gợi ý đáp án:**

- Nếu được bảng băm phục vụ cho việc tìm kiếm (0.25 điểm)
- Nếu được cách sử dụng hàm băm trong việc phân bổ giá trị (0.25 điểm)
- Nếu được ưu việt của tìm kiếm trên bảng băm so với tìm kiếm tuần tự hay nhị phân (0.25 điểm)
- Cho ví dụ minh họa được (0.25 điểm)

**HẾT**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**ĐỀ THI CUỐI KỲ MÔN CẤU TRÚC DỮ LIỆU**

**Thời gian: 90 phút**

**(Không sử dụng tài liệu)**

**Câu 1:** Trình bày kết quả thực hiện từng bước khi chạy thuật toán heapsort cho dãy số sau giảm dần: 10, 30, 20, 13, 12, 33, 6, 9, 15, 14

**Câu 2:** Giả sử cho một danh sách liên kết đơn mà mỗi phần tử trong danh sách chứa thông tin một SIÊU THỊ gồm:

THÔNG TIN SIÊU THỊ	KIỂU DỮ LIỆU
Mã siêu thị	int
Tên siêu thị	char[50]
Năm thành lập	int
Số lượng nhân viên	int

- Khai báo cấu trúc danh sách liên kết đơn để quản lý các SIÊU THỊ trên.
- Viết hàm liệt kê các siêu thị có thời gian hoạt động trên 10 năm.
- Viết hàm xóa các siêu thị có số lượng nhân viên < 100 người.

**Câu 3:** Giả sử ta có cây nhị phân tìm kiếm như sau:

- Anh (chị) hãy cho biết kết quả duyệt cây LNR, NLR.
- Anh (chị) hãy vẽ kết quả cây sau khi xóa node 36
- Viết hàm đếm cây có bao nhiêu node có 2 con.
- Viết hàm xuất tất cả các node thuộc tầng thứ K trên cây (giả sử tầng đầu tiên trên cây là tầng 1)

Ví dụ: tầng thứ 3 gồm các node:

{22, 45, 74, 93}

