

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH  
Đề 1

ĐỀ THI CUỐI HỌC KỲ II (2022-2023)  
MÔN: Cấu trúc dữ liệu & Giải thuật  
Lớp : Chính qui, Chất lượng cao  
Thời gian: 90 phút

(Sinh viên không được sử dụng tài liệu)

HỌ VÀ TÊN SV: .....	<b>CÁN BỘ COI THI</b> 2 3 4 5 6 7
MSSV: .....	
STT: .....	
PHÒNG THI: .....	

**CÂU HỎI TỰ LUẬN**

**Câu 1 (1.5 điểm) (CLO1, CLO2)**

- 1.1 Các thuật toán sắp xếp nào, trong số các thuật toán insertion sort, heap sort, merge sort và quick sort, được thiết kế theo chiến lược chia để trị? (0.5 điểm)
- 1.2 Cho dãy số  $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . Hãy cho biết:
- Những thuật toán sắp xếp nào, trong số các thuật toán heap sort, quick sort và merge sort, sẽ gặp thuận lợi khi sắp dãy số A theo thứ tự giảm dần? (0.5 điểm)
  - Thuận lợi đó, theo từng thuật toán đã trả lời ở câu 1.2 a, là gì? (0.5 điểm)

**Câu 2 (2 điểm) (CLO2, CLO3)**

Telegram đã ra đời được 10 năm và được biết đến như là một dịch vụ nhắn tin tức thời miễn phí, đa nền tảng và mã hóa. Màn hình của dịch vụ này chỉ hiển thị tin nhắn của k người sau cùng. Khi có một tin nhắn được gửi đến bạn thì tin nhắn sẽ được hiển thị vào đầu danh sách tin nhắn. Các tin nhắn sẽ được nhóm lại theo số điện thoại (SDT) của người gửi.

Cho một chuỗi tuần tự các SDT có gửi tin nhắn cho bạn:

- 2.1 Hãy mô tả cấu trúc dữ liệu (CTDL) sử dụng để hiển thị tin nhắn của k người sau cùng và các tin nhắn sẽ được nhóm lại theo SDT của người gửi (xem mô tả ví dụ input, output bên dưới) (0.5 điểm)
- 2.2 Viết chương trình bằng C++ hiện thực hoá yêu cầu, sử dụng CTDL trong câu 2.1 (1.5 điểm)

**Ví dụ Input:**

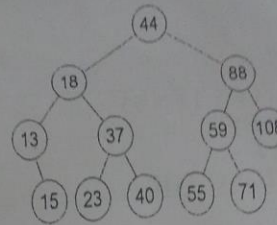
- Dòng 1: số k
  - Dòng 2: số n: số lượng tin nhắn được gửi đến bạn
  - Dòng 3: danh sách SDT của n tin nhắn theo thứ tự thời gian từ trước đến sau
- Lưu ý:** Giới hạn SDT chỉ có tối đa 3 chữ số

**Ví dụ Output:** Danh sách các SDT của các tin nhắn hiển thị trên màn hình theo thứ tự SDT có tin nhắn mới hơn sẽ xuất hiện trước kèm theo số lượng tin nhắn từ SDT đó.

Input	Output
5 12 903 901 902 904 976 976 973 986 976 904 905 986	986(2) 905 904(2) 976(3) 973

**Câu 3** (1.5 điểm) (CLO1, CLO3)

3.1 Cho dãy ký tự như sau: I, S, G, L, X, V, A, T, E, R, H, Y, hãy vẽ cây nhị phân tìm kiếm khi thêm từng ký tự vào cây theo thứ tự từ trái qua phải của dãy ký tự, biết rằng giá trị của từng ký tự tương ứng theo thứ tự xuất hiện của ký tự trong từ điển như sau: A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z (0.5 điểm)



3.2 Cho biết kết quả duyệt cây nhị phân tìm kiếm (hình bên) theo RNL, NRL (0.5 điểm)

3.3 Viết hàm đếm số nút có 2 nút con trên cây nhị phân tìm kiếm (0.5 điểm)

**Câu 4** (1.5 điểm) (CLO1)

Cho biết B-Tree bậc 5 là một cây đa nhánh thỏa mãn đồng thời tất cả các tính chất sau:

- Tất cả node lá phải nằm trên cùng một mức
- Tất cả các node, trừ node gốc và node lá có tối thiểu là 2 khóa và 3 con.
- Tất cả các node có tối đa là 4 khóa và 5 con
- Một node không phải là lá và có n khóa thì bắt buộc phải có n+1 con.

Lần lượt thêm các giá trị sau đây vào cây: 9, 8, 23, 2, 14, 17, 11, 1, 24, 16, 5, 7, 13, 4, 18, 25, 19, 22, 26, 15, 29, 3.

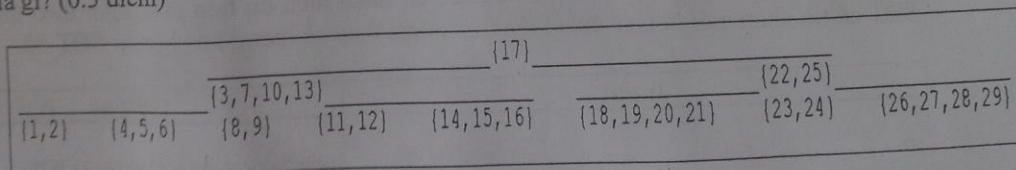
4.1 Vẽ trạng thái của cây sau khi thêm toàn bộ các giá trị trên. (1 điểm)

4.2 Cho biết cấu trúc BTree\_node và một hàm duyệt cây được định nghĩa như sau:

```
class BTree_node{
    vector<int> keys;
    vector<BTree_node*> children;
};

void dfs2(BTree_node* root){
    stack<BTree_node*> q;
    q.push(root);
    while(q.empty() != true){
        BTree_node* x = q.top(); q.pop();
        for (BTree_node* i : x->keys) cout << i << " ";
        cout << endl;
        for(int i: x->children) q.push(i);
    }
}
```

Với cây B-Tree bậc 5 có hình dáng như bên dưới thì hàm duyệt cây trên sẽ cho ra output là gì? (0.5 điểm)



**Câu 5** (1.5 điểm) (CLO1)

5.1 Cho bảng băm T có các đặc điểm sau:  
- Kích thước bảng băm m=7.



- Giá trị khóa  $k$  là các số nguyên  $k \in [1, 50]$ .
  - Sử dụng hàm băm theo phép chia.
  - Xử lý đụng độ (hay va chạm, collision) theo phương pháp thăm dò tuyến tính.
- Hãy cho biết công thức của hàm băm  $h(k)$  và hàm băm lại  $h_i(k)$ , hay  $h(k, i)$ , sao cho xác suất xảy ra đụng độ trong  $T$  không vượt quá  $n/m$  khi  $T$  có  $n$  phần tử. (0.5 điểm)

5.2 Cho bảng băm  $T$  có các đặc điểm sau:

- Giá trị khóa  $k$  là một số nguyên dương.
- Kích thước bảng băm  $m=11$ .
- Hàm băm  $h(k) = \text{floor}(((0.618 * k) \bmod 1) * m)$ .
- Hàm băm lại  $h(k, i) = \text{floor}(((0.618 * (k + i)) \bmod 1) * m)$ .

Cho biết:

- Hàm  $\text{floor}(x)$  trả về giá trị nguyên lớn nhất nhỏ hơn hoặc bằng  $x$ . Ví dụ  $\text{floor}(4.8) = 4$ .
- Biểu thức  $x \bmod 1$  sẽ lấy phần thập phân của  $x$ . Ví dụ,  $8.492 \bmod 1 = 0.492$ .
- Các số thực chỉ lấy đến ba số sau phần thập phân.

Giá sử bảng băm  $T$  đã chứa các khóa như sau:

Chỉ số	0	1	2	3	4	5	6	7	8	9	10
Khóa			2				9				

Hãy trình bày quá trình:

- Thêm khóa có giá trị 1 vào  $T$ . (0.5 điểm)
- Giá sử khóa 2 đã bị xóa (ô có chỉ số 2 trên  $T$  chứa ký hiệu DELETED). Tìm khóa có giá trị 1 trên  $T$ . (0.5 điểm)

**Câu 6 (2 điểm) (CLO2, CLO3)**

Bài toán tìm đường đi ngắn nhất giữa hai đỉnh trên đồ thị có thể được 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 và các cạnh đều có trọng số, hãy tìm một đường đi ngắn nhất (không có đỉnh lặp lại) từ đỉnh xuất phát  $s$  thuộc  $V$  đến đỉnh đích  $g$  thuộc  $V$ . Đường đi ngắn nhất giữa hai đỉnh có tổng các trọng số của các cạnh tạo nên đường đi đó là nhỏ nhất.

Giá sử thông tin đầu vào của bài toán (Input) được nhập vào chương trình và kết quả đầu ra (Output) bao gồm:

Input	Giải thích
8 10	- Dòng đầu tiên chứa hai số nguyên dương $v$ và $e$ , lần lượt là số đỉnh và số cạnh của đồ thị
ABCDEFGHIK	- Dòng tiếp theo chứa $v$ chuỗi (chuỗi không có khoảng trắng) là danh sách tên các đỉnh
AC 9	- Với $e$ dòng tiếp theo, mỗi dòng chứa hai chuỗi $u, i$ và một số nguyên dương $x$ , thể hiện thông tin có một cạnh nối từ đỉnh $u$ sang đỉnh $i$ trong đồ thị với độ dài (trọng số) là $x$
AD 14	- Dòng cuối cùng chứa hai chuỗi $s$ và $g$ , đây là đỉnh bắt đầu và đỉnh kết thúc của đường đi cần tìm
AE 3	<b>Lưu ý: thông tin nhập tương ứng với đồ thị có hình như bên dưới</b>
CH 7	
DH 1	
EI 10	
EK 15	
HK 2	
IK 11	
KB 6	
AK	
Output	Giải thích
ADHK	Danh sách các đỉnh trên đường đi (cả đỉnh xuất phát và đỉnh đích)

Hãy thực hiện các yêu cầu sau:

6.1 Xây dựng các CTDL 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.

6.2 Dưới đây là một ví dụ về mã giả của thuật toán Dijkstra để tìm một đường đi ngắn nhất từ đỉnh xuất phát s đến đỉnh đích g trong đồ thị. Hãy minh họa các bước thực hiện của thuật toán theo mã giả đã cho để "Tìm đường đi ngắn nhất từ đỉnh A đến đỉnh H" trong đồ thị bên dưới (0.75 điểm).

Gọi Open: tập các đỉnh có thể xem xét lựa chọn cho bước đi tiếp theo, các đỉnh có thể được xem xét lại, đỉnh chờ duyệt

Close: tập các đỉnh đã xét/đã duyệt, không xem xét lại

s: đỉnh xuất phát

g: đỉnh đích

p: đỉnh đang xét, đỉnh hiện hành

Hàm d(u) dùng để lưu trữ độ dài đường đi ngắn nhất từ đỉnh nguồn s đến đỉnh u

w(u,i) : trọng số của cạnh nối từ đỉnh u tới đỉnh i

parent(q)=p: lưu thông tin cha con, đỉnh p là cha của đỉnh q

Các bước thực hiện chính:

Bước 1: Khởi tạo

Open = {s}    Close = {}    d(s)=0

Bước 2: While (Open ≠ {})

2.1 Chọn p thuộc Open có d(p) nhỏ nhất (xóa p ra khỏi Open). Nếu có nhiều đỉnh cùng giá trị d(p) nhỏ nhất thì chọn đỉnh nhập vào sau trong danh sách đỉnh

2.2 Nếu p là đỉnh đích thì xuất đường đi, kết thúc thuật toán

2.3 Nếu p đã duyệt rồi thì bỏ qua, không xem xét lại, trở lại đầu vòng lặp

2.4 Đánh dấu p đã duyệt qua rồi (tức thêm p vào Close)

2.5 Với mỗi đỉnh q kề với p, nếu q không thuộc Close:

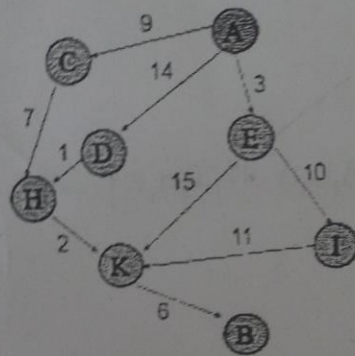
2.5.1 Nếu q đã có trong Open và  $d(q) > d(p) + w(p,q)$  thì cập nhật các thông tin:

$d(q) = d(p) + w(p,q)$     parent(q)=p

2.5.2 Nếu q chưa có trong Open:

$d(q) = d(p) + w(p,q)$     parent(q)=p    Thêm q vào Open

Bước 3: Kết luận "Không tìm thấy đường đi"



6.3 Hãy thiết kế CTDL cho các đối tượng được đề cập trong thuật toán 6.2 (như Open, Close, hàm  $d(u)$ , parent) và viết hoàn chỉnh hàm Dijkstra để giải quyết bài toán nêu trên (0.75 điểm).

Hết