

# CẤU TRÚC DỮ LIỆU & GT

## ÔN THI CUỐI KỲ

---

GIẢNG VIÊN: THS NGUYỄN BÁCH VÂN

1

## THÔNG TIN

---

Thời gian thi: 90 phút

Ngày thi: ca 2 ngày 28.6.2022

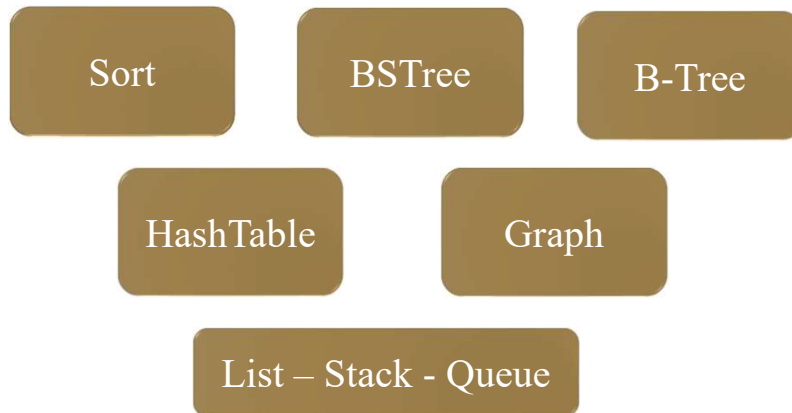
Số lượng đề: 01

Ghi chú: Không tham khảo tài liệu

2

## NỘI DUNG

---



3

## 1. Sort

---

Các thuật toán sắp xếp: **selection, insertion, quick, heap, merge, radix, counting**

- Ý tưởng, độ phức tạp  $O()$
- Các bước thuật toán, code
- Chạy tay trên dãy cụ thể: dãy số hoặc ký tự, sắp xếp tăng hoặc giảm

4

## 1. Sort

---

### Câu hỏi: 2 điểm

- Hãy cho biết **độ phức tạp** của thuật toán **Quick sort** theo định nghĩa Big-O (O lớn) (0.25 điểm)
- Viết **hàm** sắp xếp mảng 1 chiều **giảm** dần với thuật toán **Quick 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)

5

## 2. List – Stack – Queue

---

Các thao tác trên List – Stack – Queue

- Khai báo cấu trúc node, List – Stack – Queue
- Tạo node CreateNode có info = x
- Thêm 1 node vào List – Stack – Queue  
(AddHead, AddTail, AddAfterQ, Push, EnQueue)
- Xóa node ra khỏi List – Stack – Queue  
(RemoveHead, RemoveAfterQ, RemoveX, Pop, DeQueue)
- In / tìm kiếm / sắp xếp trên List – Stack – Queue

6

### 3. BSTree

---

Cho dãy số hoặc ký tự

- Vẽ cây: vẽ 1 cây kết quả
- Xóa node: vẽ từng cây sau khi xóa 1 node, cây xóa node sau sẽ vẽ từ cây kết quả xóa node trước đó
- Duyệt cây: duyệt theo thứ tự đề yêu cầu
- Viết hàm: các bài tập wecode

7

### 3. BSTree

---

#### Câu hỏi:

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

Hãy thực hiện các yêu cầu sau:

- a. **Vẽ cây** NPTK bằng cách thêm lần lượt từng ký tự trên. (1 điểm)
- b. Cho biết kết quả **duyet cây** theo RNL, NRL. (0.5 điểm)
- c. **Hủy** lần lượt từng nút D, E, F, H trên cây, mỗi lần hủy 1 nút vẽ lại cây nối tiếp theo như thứ tự hủy. (1 điểm)
- d. Viết **hàm** đếm số lượng nút có một nút con trên cây, nếu cây rỗng thì in ra giá trị -1. (1 điểm)

8

### 3. BSTree

**Câu hỏi:** Viết hàm **XUẤT/ĐẾM/TÍNH TỔNG** các node có đặc điểm đặc biệt (số chẵn/lẻ/số nguyên tố/số đối xứng/số tăng giảm... HOẶC node có 1 con /2 con/node lá...)

```
void DieuKienNode(Tree T, int &Dem, int &S)
{
    if (T)
    {
        if (T->left==NULL && T->right==NULL)
        {
            cout << T->key << " "; //Xuất
            Dem++; //Đếm
            S += T->key; //Tổng
        }
        DieuKienNode(T->left, Dem, S);
        DieuKienNode(T->right, Dem, S);
    }
}
```

**Node lá:**  
if (T->left==NULL && T->right==NULL)  
**Node 1 con:**  
if (T->left!=NULL && T->right==NULL ||  
T->left==NULL && T->right!=NULL)  
**Node có giá trị chẵn:**  
if (T->key%2==0)  
**Node có giá trị là số nguyên tố:**  
if (KtraSNT(T->key)==true)

9

### 4. B-Tree

Cho cây B-Tree bậc 3 / bậc 5 hoặc bậc lớn hơn

- Vẽ cây: vẽ riêng 1 cây cho các khóa không làm split node và 1 cây cho 1 khóa làm split node
- Xóa khóa: vẽ từng cây sau khi xóa 1 khóa, cây xóa khóa sau sẽ vẽ từ cây kết quả xóa khóa trước đó

10

## 4. B-Tree

---

Bậc  $m=2*k+1$

- Tối đa:  $2*k$  khóa / node
- Tối thiểu:  $k$  khóa / node

Bậc 5

- Tối đa: 4 khóa / node
- Tối thiểu: 2 khóa / node

Bậc 3

- Tối đa: 2 khóa / node
- Tối thiểu: 1 khóa / node

11

## 4. B-Tree

---

**Thêm** node tuân thủ theo qui tắc sau:

- Thêm khóa cần sắp xếp theo thứ tự tăng dần từ trái qua phải
- Nếu khóa cần thêm vào làm vi phạm điều kiện tối đa thì **tách (slipt) node** → chọn khóa ở giữa đẩy lên node cha

12

## 4. B-Tree

- Xóa khóa ở node lá không làm vi phạm đk tối thiểu
- Xóa khóa ở node lá làm vi phạm đk tối thiểu
  - Underflow**: khi xóa khóa làm node vi phạm mà node kế bên có số lượng khóa nhiều hơn tối thiểu --> mượn khóa thay thế --> không làm thay đổi số khóa của node cha
  - Catenation**: khi xóa khóa làm 2 node kế nhau có số lượng khóa  $< 2*k$   
--> gộp node --> làm node cha bị mất khóa
- Xóa khóa ở node **khác** lá --> tìm khóa ở node lá thay thế (**max\_left** / min right)--> xử lý giống trường hợp xóa khóa ở node lá như trên

13

## 4. B-Tree

### Câu hỏi:

Cho dãy số: 27, 3, 28, 33, 8, 19, 23, 9, 1, 30, 11, 21, 5, 13, 2, 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 5 rồi thì:

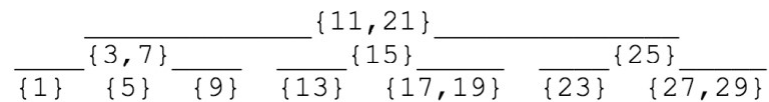
- Các khóa nào khi thêm vào sẽ làm phát sinh thao tác **split** node? (0.5 điểm)
- Vẽ cây B-Tree trước và sau khi thêm các khóa trên. (1 điểm)

14

## 4. B-Tree

### Câu hỏi:

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)

15

## 4. B-Tree

### Câu hỏi:

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

Hỏi khi lần lượt thêm các số trong dãy theo thứ tự từ trái qua phải vào một cây B-Tree bậc 5 rồi 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)

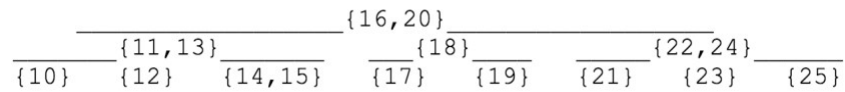
16



## 4. B-Tree

### Câu hỏi:

Cho cây B-Tree bậc 3 như hình sau:



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

17

## 5. HashTable

- Liệt kê các dạng hàm băm đã học (có công thức của các dạng hàm băm)
- Hãy trình bày các phương pháp xử lý đụng độ (có công thức)
- Hãy trình bày Input / Output / thuật toán của các thao tác tìm kiếm, thêm, xóa phần tử trong bảng băm.

18

## 5. HashTable

### Câu hỏi:

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)  $\text{prob}(\text{key}, i)$  như sau:

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

Cho  $M = 7$  và trên bảng băm đã chứa các mục dữ liệu như bên dưới.

0	1	2	3	4	5	6
		2		4		

Trình bày từng bước việc **Thêm/Tìm/Xóa** các khóa  $\text{Key} = \{6, 16, 10\}$  vào bảng băm theo đúng thứ tự trong danh sách.

19

## 6. Graph

Cho bài toán cụ thể và yêu cầu:

- Xây dựng cấu trúc dữ liệu đồ thị để lưu trữ (ma trận kề hoặc danh sách kề)
- Viết code các thuật toán duyệt đồ thị DFS / BFS hoặc tìm đường đi ngắn nhất giữa 2 đỉnh bằng thuật toán Dijkstra

20

## NỘI DUNG

### Câu hỏi:

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.

21

## NỘI DUNG

### Câu hỏi:

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 $e$ là số cạnh của đồ thị
Viet_Nam Lao	- $e$ dòng tiếp theo, mỗi dòng chứa 02 chuỗi $u$ và $i$ , thể hiện thông tin có một cạnh nối từ đỉnh $u$ sang đỉnh $i$ trong đồ thị
Viet_Nam Trung_Quoc	
Thai_Lan Lao	
...	Lưu ý: không biết trước số đỉnh và danh sách các đỉnh
Campuchia Thai_Lan	

22

## NỘI DUNG

---

### Câu hỏi:

Hãy thực hiện các yêu cầu sau:

- a. 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)
- b. 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)