

ĐỀ THI CUỐI KÌ 21CTT3

Thời gian làm bài: 75 phút

Dữ liệu sử dụng trong đề bài là dữ liệu từ điển Anh - Việt về thú vật dành cho trẻ em có nội dung như sau:

```
284 5
285 Piglet
286 ('pɪɡlət)
287 Lợn con
288 1
289 Plaice
290 /pleɪs/
291 Cá bơn
292 5
293 Polar bear
294 ('pəʊlə beə)
295 Con gấu Bắc cực
296 2
297 Porcupine
298 ('pɔ:kjʊpaɪn)
299 Con nhím.
300 2
301 Porcupine
302 ('pɔ:kjʊpaɪn)
303 Con nhím
304 2
305 Praying mantis
306 ('preɪɪŋ 'mæntɪs)|
307 Bọ ngựa
```

Cho các định nghĩa struct sau:

```
struct Animal
{
    string en;
    string vn;
    string ph;
    int loai;
};
```

```
struct AVLNode
{
    int key;
    Animal data;
    Node* pLeft;
    Node* pRight;
};
```

Thực hiện các yêu cầu sau:

Câu 1 (6 điểm)

1. (3 điểm) Để cho phép việc tìm kiếm từ điển có chi phí tối đa là $\log(n)$. Hãy đọc danh sách thú vật và lưu vào cây AVL (dựa trên thuộc tính `key`). Biết rằng thuộc tính `key` trong struct `AVLNode` có giá trị là: $\sum_{i=0}^n(s[i])$ với $s[i]$ là kí tự thứ i trong thuộc tính `en`. Lưu ý: Khóa `key` trùng thì không thêm vào.

- `AVLNode* readAnimals(string filename);`
- Input: `filename` - tệp tin dữ liệu `"data.txt"`

2. (2 điểm) Cài đặt hàm cho phép người dùng xóa một con vật ra khỏi cây AVL

- `void removeNode(AVL* &root, string en)`

3. (1 điểm) Cài hàm cho phép in ra danh sách từ điển theo mức từ cây AVL

- `void levelOrder(AVL* root)`

Câu 2 (4 điểm)

Cho một đồ thị liên thông, vô hướng, có dạng cây gồm n đỉnh - được đánh số từ 0 tới $n-1$. Cho trước một số nguyên n - là số đỉnh và một ma trận có e dòng 2 cột trong đó $e[i] = [a, b]$ nghĩa là có một cạnh nối giữa a và b trong đồ thị trên.

- 1. (1 điểm) Hãy biểu diễn lại đồ thị trên dưới dạng ma trận kề.

– `int** convertAdjacencyMatrix(vector<vector<int>> edges, int n)`

– Ví dụ:

* Input:

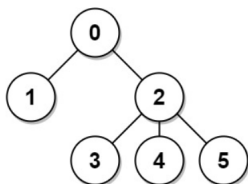
```
0 1
0 2
1 3
```

* Output:

```
0 1 1 0
1 0 0 1
1 0 0 0
0 1 0 0
```

- 2. (3 điểm) Hãy trả về mảng `answer` gồm n phần tử, trong đó `answer[i]` = tổng khoảng cách từ node thứ i đến tất cả các node còn lại.

– Ví dụ:



* Input: $n = 6$, `edges = [[0,1],[0,2],[2,3],[2,4],[2,5]]`

* Output: `[8,12,6,10,10,10]`

* Giải thích: Với node số 0, tổng khoảng cách = k/c giữa (0,1) + k/c giữa (0,2) + k/c giữa (0,3) + k/c giữa (0,4) + k/c giữa (0,5) = $1 + 1 + 2 + 2 + 2 = 8$. Tương tự với node 1, 2, 3, 4, 5