**Big Update - ngày 29/11**

**bởi**[**Băng Ngọc Bảo Tâm**](https://e-learning.hcmut.edu.vn/user/view.php?id=99991&course=116047)**- Thứ tư, 29 Tháng mười một 2023, 9:24 PM**

**Number of replies: 13**

**Một vài thay đổi và chỉnh sửa để làm rõ hơn yêu cầu của bài tập lớn như sau (bản pdf update sẽ có trong cuối tuần)**

* **Chuẩn hóa tên khách hàng - Lệnh LAPSE**

**- Quy trình chuẩn hóa hiện tại phải cần đến 2 lần sort (1 lần sau khi tính freq và 1 lần sau khi cộng dồn) -> Việc này là dư thừa bởi vì chúng ta chỉ cần 1 lần sort sau khi cộng dồn là được. Cụ thể, sau khi các em tạo ra được list freq của các ký tự, các em tiến hành cộng dồn lại (lúc này không cần quan tâm thứ tự). Sau khi cộng dồn, các em sort lại một lần theo thứ tự alphabet (ký tự viết thường > ký tự viết hoa, có nghĩa là a..z > A..Z) là các em sẽ được 1 danh sách dùng để build cây huffman. Ví dụ: "abaaabbbDd" thì đầu tiên ta được a:4 -> b:4 -> D:1 -> d:1, mã hóa ta được e:4 -> f:4 -> E:1 -> e:1, cộng dồn được: e:5 ->f:4 -> E:1, sort lại ta được E:1 -> f:4 -> e:5, lấy list này build cây.**

**- Trong quá trình build cây huffman, nút nào được đưa vào danh sách trước sẽ được xử lý trước. Ví dụ list a:3 -> b:3 -> c:5 ->d:6 thì xem như nút a vào trước b, b trước c, c trước d. Sau step đầu (gộp a và b) thì list mới sẽ như sau: c:5 ->d:6 -> (ab:6). Vì nút gộp ab mới được tạo ra trong khi d đã được đưa vào trước. Cứ lặp lại quá trình này đến khi build thành công cây huffman**

**- Việc xoay cây theo cơ chế AVL, các em chỉ cần quan tâm đến cơ chế xoay (xét balance, xoay LL, RR, LR, RL), cứ xoay từ trên xuống cho đến khi cây được balance. Việc mất cân bằng dạng 3-1 hay 4-1 các em không cần quan tâm, vì bản chất nó đâu phải là AVL, cứ coi như đây là 1 task là được. Nếu trong lúc xoay có trường hợp nút root là ký tự thì xem như khách hàng ảo, loại khách hàng này. Qua xử lý lệnh kế tiếp.**

**- Sau khi đã build cây xong, ví dụ tên khách hàng các em mã hóa được 110101010110101011 thì chuỗi 10 ký tự lấy từ phải sang trái là 1101010110. Nếu chuỗi ít hơn 10 ký tự thì cứ lấy ít hơn.**

* **Bố trí khách ở nhà hàng Phục ma ngự - Lệnh LAPSE**

**- NUM biểu thị cho số khách hàng tại khu vực. Trường hợp swap giữa nút cha và nút con, sẽ chọn nút con có giá trị nhỏ hơn để swap. Ngoài ra, nếu các khu vực có cùng NUM thì khu vực nào được thêm vào HEAP sớm hơn sẽ được xem như nhỏ hơn.**

* **Lệnh KUKUSEN:**

**- Vì số lượng các hoán vị có thể rất lớn, dẫn đến trường hợp xóa cả cây, cho nên khi tính ra được số hoán vị Y, thì tiếp tục lấy Y = Y % MAXSIZE và tính toán tiếp.**

* **Lệnh KEITEIKEN:**

**- Về việc đuổi NUM khách ở NUM khu vực, ví dụ NUM bằng 2, thì các em tìm 2 khu vực thỏa yêu cầu đề bài. Sau đó tại mỗi khu vực, đuổi 2 khách. Xử lý khu vực nào xong, các em cập nhật ngay trong cây HEAP rồi mới xử lý khu vực tiếp theo.**

* **Lệnh HAND:**

**- Các em sẽ in cây huffman theo thứ tự tự in-order theo format "data/n", với data là giá trị của node trong cây (ký tự hoặc số freq). Và chỉ cần xét đến khách hàng nào vừa đến 1 trong 2 nhà hàng gần đây nhất, không cần quan tâm đến việc người đó bị đuổi về hay còn ở nhà hàng.**

* **Một số vấn đề khác:**

**- Trong quá trình xoay cây AVL sẽ có trường hợp (A (B(C D))), trường hợp này để mình thống nhất với các thầy về cách xoay. Nhưng chắc chắn sẽ dựa vào hệ số cân bằng tại nút đó.**

**- Trong bài tập lớn này, khẩu hiệu "thiên thượng thiên hạ, duy ngã độc tôn" sẽ đổi  thành "thiên thượng thiên hạ, duy ngã đọc tên" cho nên khách hàng có cùng tên với nhau vào nhà hàng cũng không thành vấn đề gì.**

**- Ngoài ra, các em được phép dùng tất cả những gì các em có để giải quyết bài tập lớn lần này.**

**Làm rõ các vấn đề trong threat Big Update**

**bởi**[**Băng Ngọc Bảo Tâm**](https://e-learning.hcmut.edu.vn/user/view.php?id=99991&course=116047)**- Chủ nhật, 3 Tháng mười hai 2023, 9:50 AM**

**Number of replies: 10**

**Mình thông tin thêm đến các em để các em rõ hơn cách hiện thực trong bài tập lớn:**

* **Thứ tự các char freq khi build cây huffman**

**- Mình có đề cập đến nội dung a..z > A..Z ý mình ở đây là thứ tự trước sau của các ký tự viết thường sẽ được xếp trước các ký tự viết hoa. Cụ thể, khi có trường hợp bằng freq, xét theo thứ tự alphabet chữ viết thường trước, sau đó mới xét thứ tự alphabet của chữ viết hoa. Ví dụ: a : 1 -> d : 1 -> A : 1 -> B : 1 -> D : 1 -> c : 2 -> C : 2 -> D : 3.**

**- Trường hợp cây chỉ có 1 nút thì sẽ có giá trị là 0.**

* **Về vấn đề xoay cây AVL**

**- Mình thấy một số em chưa đưa ra được ví dụ trong một vài trường hợp mất cân bằng mà cứ thắc mắc nếu gặp trường hợp đó thì giải quyết làm sao nên bây giờ mình tinh gọn việc xoay cây AVL như thế này. Bây giờ các em không cần xoay đến khi nào đạt được trạng thái cân bằng nữa, mà các em chỉ cần xoay tối đa 3 lần. Ở mỗi lần xoay, các em sẽ chọn node mất cân bằng để tiến hành xoay, xét theo thứ tự pre-order (NLR) tính từ root để các em tiến hành xoay tiếp. Khi hết 3 lần xoay thì các em dừng và lưu lại cây kết quả. Trường hợp nếu như các em xoay ít hơn 3 lần mà đã đạt được trạng thái cần bằng thì thôi.**

**- Khi gặp trường hợp A(B (C (X X) D(Y Y) E) hoặc A (B (C D)), hoặc các trường hợp tương tự (có 1 nhánh full node), thì trong bài tập lớn này, để đơn giản các em chỉ cần xoay 1 lần (LL hoặc RR tùy trường hợp) là được.**

**- Nếu như các em có thắc mắc về 1 trường hợp xoay cây nào các em chưa rõ thì các em vẽ cái hình rồi upload lên trên forum để mọi người cùng vào xem và giải đáp nhé.**

* **Ví dụ hàm HAND**

**Giả sử cây huffman của vị khách vừa vào nhà hàng là (30 (a 15 (B C)) thì kết quả in ra màn hình là: a/n30/nB/n15/nC/n**