Cấu Trúc Dữ Liệu Và Giải Thuật

Bài tập lớn 1

JJK RESTAURANT OPERATIONS

nhóm thảo luận CSE
https://www.facebook.com/groups/211867931379013

Tp. Hồ Chí Minh, Tháng 6/2023

Mục lục

1 Kiên thức cần có

- Danh sách liên kết đôi
- Danh sách liên kết vòng
- Sell sort

```
class customer {
public:
    string name;
    int energy;
    customer* prev;
    customer* next;
    friend class Restaurant;
public:
    customer(){}
    customer(string na, int e, customer* p, customer *ne)
    : name(na), energy(e), prev(p), next(ne){}
    ~customer(){
        delete prev;
        delete next;
    }
    void print() {
        cout << name << "-" << energy << endl;
    }
};
```

- *NAME*: một chuỗi ký tự trong bảng chữ cái Alphabet (bao gồm cả chữ viết thường và viết hoa) liên tục không có khoảng trắng, biểu thị tên của khách hàng.
- *ENERGY*: một số nguyên biểu thị năng lượng của các chú thuật sư (giá trị dương), và oán linh (giá trị âm).
- *NUM*: một số nguyên với ý nghĩa khác nhau ứng với từng lệnh xử lý khác nhau. Và ứng với mỗi lệnh thì giá trị NUM này sẽ có các khoảng giá trị khác nhau.
- *MAXSIZE*: số lượng bàn tối đa mà nhà hàng có thể phục vụ. Tuy nhiên giá trị này có thể tạm thời thay đổi trong quá trình vận hành nhà hàng.

```
private:
    customer * customerX; //! lưu trữ danh sách khách hàng đang ở trong bàn
    int sizeCustomerInDesk;
    customer * customerQueue; //! mô tả danh sách khác hàng đang trong hàng đợi chỉ sử dung ne
    int sizeCustomerQueue;
```

- *customerX* khách hàng được vị trí gần nhất vừa có sự thay đổi (ngồi vào bàn hoặc ra khỏi bàn). **Được biểu diễn Danh sách liên kết vòng kép Doubly Circular linked list**
- *sizeCustomerInDesk* số lượng khách hàng đang có trong bàn
- *customerQueue* danh sách khách hàng đang chờ **Được biểu diễn Danh sách liên kết vòng Circular linked list**
- *sizeCustomerQueue* số lượng khách hàng đang chờ
- Để tiện nên sử dụng chung *class customer* nên *customerQueue* không dùng *prev*

2 Task1 RED

2.1 Hiện Thực

Mô tả: Hàm này có chứ năng là sắp xếp vị trí chỗ ngồi cho khách. Thực hiện theo các bước sau

1. Bước 1: Nhà hàng chỉ tiếp đón chú thuật sư (ENERGY dương) hoặc oán linh (ENERGY âm) và từ chối nhận khách hàng có ENERGY bằng 0. -> ENERGY bằng 0 đuổi khách về
2. Bước 2: Hàng chờ đã đạt đến số lượng tối đa thì nhà hàng sẽ ngưng nhận khách. -> Hàng chờ đầy đuổi khách về
3. Bước 3: Và bởi vì "thiên thượng thiên hạ, duy ngã độc tôn", nên nhà hàng sẽ không đón tiếp các vị khách đến sau và không cho phép vào hàng chờ nếu có tên giống với tên của các vị khách đang dùng bữa trong nhà hàng (hoặc đã có tên trong hàng chờ). Ví dụ một chú thuật sư tên ABC đang dùng bữa thì những chú thuật sư hoặc oán linh đến sau có tên ABC sẽ bị mời về. -> xét thử khách hàng trong hàng chờ và trong bàn có chung tên không, nếu chung thì đuổi khách về
4. Bước 4: Nếu số lượng khách ngồi vào bàn đã đạt đến MAXSIZE thì sẽ dừng việc nhận khách và đưa khách vào hàng chờ
   (a) Bước 4.1: xét xem trong hàng chờ có người nào không -> chèn trường hợp size=0
   (b) Bước 4.2: Thêm khách hàng vào cuối hàng chờ
5. Bước 5: Khách hàng đầu tiên vào quán -> Thêm khách hàng vào danh sách và cập nhật customerX và size
6. Bước 6: Tuy nhiên, khi số lượng khách trong bàn lớn hơn hoặc bằng MAXSIZE/2. Nhân viên sẽ đổi chiến lược chọn chỗ ngồi cho khách. Vì biết những người có ENERGY gần bằng nhau thường không thích ngồi gần nhau. Do đó trước khi chọn vị trí, nhân viên sẽ tính giá trị chênh lệch lớn nhất giữa vị khách mới với tất cả vị khách trong nhà hàng thông qua việc lấy hiệu trị tuyệt đối của từng cặp ENERGY ứng với từng khách hàng (giả sử gọi là RES) để quyết định chỗ ngồi. -> Nếu vào trường hợp này cần tìm vị trí X mới
7. Bước 7: Khi vị khách tiếp theo đến, nhân viên sẽ bố trí chỗ ngồi cho khách tại vị trí liền kề bên phía thuận chiều kim đồng hồ nếu ENERGY của khách lớn hơn hoặc bằng với ENERGY của khách tại vị trí X. Ngược lại, khách sẽ ngồi tại vị trí liến kề bên phía ngược chiều kim đồng hồ của khách tại vị trí X -> Tạo Thông tin khách hàng mới sau đó cập nhật next prev của khách hàng mới khi thêm vào và cập nhật khác hàng trước khác hàng mới và sau khách hàng mới. Cập nhật CustomerX và size

2.2 Test Case

1. Test 1: xét bước 1, bước 3, bước 5.
2. Test 2: Chèn theo chiều kim đồng hồ của Bước 7 và print theo chiều kim đồng hồ
3. Test 3: Chèn theo ngược đồng hồ của Bước 7 và print theo chiều kim đồng hồ
4. Test 4: Chèn theo chiều kim đồng hồ của Bước 7 và print theo ngược chiều kim đồng hồ
5. Test 5: Chèn theo ngược đồng hồ của Bước 7 và print theo ngược chiều kim đồng hồ
6. Test 6: xét Bước 6 với MAXSIZE chẵn
7. Test 7: xét Bước 7 với MAXSIZE lẻ
8. Test 8: xét bước 4, bước 3, bước 2.
9. Test 9 -> 100: random.

3 Hướng dẫn chạy test case

1. Yêu cầu cài $g++$, IDE Vscode (Vs có nhiều ràng buộc về code nên ít sài sài Vscode cho dễ), cài extensions vscode cài better comment
2. Tải từng task về tại đây **BTL**
3. Mở *new Termial* lên trong vscode
4. gõ lệnh g++ -o main main.cpp sau đó
   (a) Nếu muốn chạy hết test case gõ lênh .\main all
   (b) Nếu muốn chạy từng test case gõ lênh .\main number với number là test case luôn chạy
   (c) Nếu muốn chạy trong 1 đoạn từ [i,j] gõ lênh .\main i j
5. Sẽ có video hương dẫn kĩ hơn