Hashing e Heap

Monitoria Algoritmos 2020.3 - Lista 3

OBS: Não é permitido o uso de tabelas hash prontas (dicionários em Python, HashMap em Java, etc.), você deverá implementá-la para a questão. Você deve implementar a estrutura Heap, seja mínimo ou máximo, para as filas de prioridade da questão (Cada fileira de assentos e a "fila" de pessoas sem assento, pelo menos).

Em um teatro, quando ocorrem eventos com entrada gratuita, a demanda de pessoas geralmente é maior do que sua capacidade. Para resolver este problema, o teatro adotou um sistema de cadastro online para tais eventos.

Ao se cadastrar, cada pessoa tem uma prioridade P no sistema, que é determinada com base nos dados fornecidos.

O teatro tem uma quantidade de fileiras F e uma quantidade Q de cadeiras por fileira. Quando uma pessoa é cadastrada para o evento, podem ocorrer as seguintes situações:

1. Caso tenha assento disponível, esta pessoa é alocada na primeira fileira com assento disponível (respeitando a ordem f1 -> fF, sendo f1 a primeira fileira e fF a última).

2. Caso todos os assentos estejam ocupados: 2.1. Sendo P1 a pessoa com menor prioridade dentre todas as sentadas, e fP1 sendo a fileira em que P1 está sentada, Se a pessoa cadastrada tiver prioridade maior do que P1, esta pessoa passa a ocupar algum assento na fileira fP1, e P1 passa a assistir o evento sem assento. 2.2. Se esta pessoa não tem uma prioridade maior do que alguém que esteja ocupando um assento, esta pessoa assistirá o evento sem assento.

Uma pessoa pode ser removida do evento caso solicitado, podendo ocorrer os seguintes cenários: Caso ela esteja ocupando um assento, ela deverá ser retirada do evento, e a pessoa com a maior prioridade dentre as que estão sem assento, passa a ocupar algum assento na fileira da pessoa removida.

Caso ela não esteja ocupando assento, ela simplesmente será retirada do evento (retirada da "fila" de pessoas sem assento).

Ao ser cadastrada, cada pessoa recebe um número de cadastro com base na ordem do cadastro, ou seja, a primeira pessoa a ser cadastrada tem número de cadastro 1, a segunda pessoa cadastrada tem número de cadastro 2, etc.

**OBS: Em casos de prioridades iguais, quem foi cadastrado primeiro terá maior prioridade nestes casos. **

O código deve ter uma tabela hash que será utilizada nos comandos de entrada "VER". Cada pessoa cadastrada é identificada unicamente através de seu **nome e número de cadastro.**

Input Specification

por espaço.

> **F Q**

*OBS: Nos casos de teste, a quantidade de pessoas totais (sentadas + sem assento) nunca irá ultrapassar 2(F*Q).**

Em seguida, será dado um inteiro **N**, correspondente à quantidade de comandos que serão dados ao sistema.

> **N**

Em seguida, serão dados N comandos, que podem ser:

- CAD X P -> Cadastra a pessoa com nome X e prioridade P.
- REM X C -> Remove a pessoa de nome X e número de cadastro C do evento.
- VER X C -> Verifica a situação da pessoa de nome X e número de cadastro C.

Output Specification

Para cada comando “CAD”, deve ser impresso:

X (C) foi alocado(a) na fileira f -> Caso a pessoa foi elegível para ocupar algum assento, deve-se imprimir esta mensagem com X sendo o nome da pessoa, C sendo o seu número de cadastro e f sendo o número da fileira em que ela foi inserida.

X (C) nao foi alocado(a) em nenhuma fileira -> Caso a pessoa cadastrada não seja elegível para ocupar um assento, conforme explicado acima. (Assistirá o evento sem assento até então)

Para cada comando “REM”, deve ser impresso:

Removido(a) -> Caso a pessoa seja encontrada e removida do evento **Inexistente** -> Caso a pessoa não seja encontrada na lista de cadastrados do evento

Para cada comando “VER”, deve ser impresso:

Sem assento -> Caso a pessoa verificada não esteja alocada em algum assento de alguma fileira.

Sentado(a) na fileira f -> Caso a pessoa verificada esteja ocupando algum assento, deve ser impressa esta mensagem com “f” sendo o número da fileira que ela está.

Inexistente -> Caso a pessoa não esteja cadastrada no evento

Sample Input #1

```
1 8
16
CAD Marina 81
CAD Andre 37
CAD Francisco 65
CAD Kaique 76
```

Sample Output #1

```
Marina (1) foi alocado(a) na fi
Andre (2) foi alocado(a) na fil
Francisco (3) foi alocado(a) na
Kaique (4) foi alocado(a) na fi
Bruna (5) foi alocado(a) na fil
Danilo (6) foi alocado(a) na fi
```

```
3   CAD Carolina 58
4   CAD Fernando 45
5   CAD Bruno 54
6   CAD Rodrigo 10
```

```
3   Bruno (3) foi alocado(a) na fil
4   Rodrigo (4) foi alocado(a) na f
5   Gabriela (5) foi alocado(a) na
6   Ian (6) foi alocado(a) na filei
```

Sample Input #3

```
1   2 2
2   23
3   CAD Eraldo 40
4   VER Eraldo 1
5   CAD Wilson 30
6   VER Wilson 2
```

Sample Output #3

```
1   Eraldo (1) foi alocado(a) na fi
2   Sentado(a) na fileira 1
3   Wilson (2) foi alocado(a) na fi
4   Sentado(a) na fileira 1
5   Guilherme (3) foi alocado(a) na
6   Pedro (4) foi alocado(a) na fil
```