

MC202 — ESTRUTURAS DE DADOS

Laboratório 06 — Filas de espera

Tarefa

Jessica é gerente de atendimento de uma empresa. Nessa empresa, o atendimento é feito em fila única e um número de chegada é dado para o cliente ao chegar na fila. Também, a empresa tem k categorias de prioridade onde $k > 0$. As prioridades são numeradas de 0 a $k - 1$ e serão atendidos primeiro os clientes com maiores prioridades. Entre os clientes que têm prioridades iguais, serão atendidos primeiro os com menores números de chegada.

Jessica relatou o seguinte problema no atendimento da empresa: às vezes, quando acaba o expediente de um atendente, os clientes ficam perdidos no saguão de atendimento e acabam se misturando. Jessica te contratou para construir um programa que receba um conjunto de clientes perdidos no saguão não necessariamente ordenados com suas respectivas prioridades e número de chegada e crie uma única fila (ordenada corretamente) com todos os clientes.

Para deixar seu programa mais rápido, Jéssica informou que os clientes estão bem distribuídos entre as k categorias. Isto é, organizar os clientes pela prioridade pode deixar seu programa mais rápido.

Entrada

A entrada é composta por um inteiro k ($1 \leq k \leq 10^4$) que é o número de prioridades, seguido por um inteiro N ($0 \leq N \leq 10^5$) que é o número de clientes, seguido por N linhas contendo um inteiro c_i e um inteiro p_i que são respectivamente o número de chegada e o número da prioridade do cliente i .

Saída

N linhas com o número de chegada e a prioridade de todos os clientes em uma única fila ordenada.

Exemplo

Entrada

2
6
7 1

```
2 0
5 0
7 0
1 1
2 0
```

Saída

```
1 1
7 1
2 0
2 0
5 0
7 0
```

Entrada

```
2
4
1 1
3 1
2 0
4 0
```

Saída

```
1 1
3 1
2 0
4 0
```

Entrada

```
10
10
0 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
```

9 9

Saída

```
9 9
8 8
7 7
6 6
5 5
4 4
3 3
2 2
1 1
0 0
```

Critérios específicos

- Você deverá implementar e utilizar uma estrutura de dados do tipo lista ligada para armazenar os clientes
- Sua implementação deve obrigatoriamente utilizar o fato de que os clientes estão bem distribuídos entre as categorias para obter uma implementação eficiente
- Deverão ser submetidos os seguintes arquivos:
 - lista.h (sua interface da estrutura lista)
 - lista.c (implementação da estrutura lista)
 - lab06.c (solução do problema do laboratório)
- Tempo máximo de execução: 1 segundo.

Testando

Para compilar usando o Makefile fornecido e verificar se a solução está correta basta seguir o exemplo abaixo.

```
make
./lab03 < arq01.in > arq01.out
diff arq01.out arq01.res
```

onde arq01.in é a entrada (casos de testes disponíveis no SuSy) e arq01.out é a saída do seu programa. O Makefile também contém uma regra para testar todos os testes de uma vez; nesse caso, basta digitar:

```
make testar_tudo
```

Isso testará o seu programa com os casos abertos. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados digitando-se:

```
make baixar_fechados
```

Para ver quanto tempo seu programa demora em cada teste, digite:

```
make tempo
```

Observações gerais

No SuSy, haverá 3 tipos de tarefas com siglas diferentes para cada laboratório de programação. Todas possuirão os mesmos casos de teste. As siglas são:

1. **DRAFT:** Esta tarefa serve para testar o programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém arquivos submetidos aqui não serão corrigidos.
2. **ENTREGA:** Esta tarefa tem limite de *uma única submissão* e serve para entregar a versão final dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa: submeta aqui quando não for mais fazer alterações no seu programa.
3. **FORAPRAZO:** Esta tarefa tem limite de *uma única submissão* e serve para entregar a versão final após o prazo estabelecido para o laboratório, mas com nota reduzida (conforme a ementa). O envio nesta tarefa irá substituir a nota obtida na tarefa ENTREGA apenas se o aluno tiver realizado as correções sugeridas no feedback ou caso não tenha enviado anteriormente em ENTREGA.

Observações sobre SuSy:

- Versão do GCC: gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
- Flags de compilação:
-std=c99 -Wall -Werror -Werror=vla -pedantic-errors -g -lm

Além das observações acima, esse laboratório será avaliado pelos critérios gerais:

- Indentação de código e outras boas práticas, tais como:
 - uso de comentários (úteis e apenas quando forem relevantes);
 - código simples e fácil de entender;
 - sem duplicidade (partes que fazem a mesma coisa).
- Organização do código:

- tipos de dados criados pelo usuário e funções bem definidas e tão independentes quanto possível.
- Corretude do programa:
 - programa correto e implementado conforme solicitado no enunciado;
 - inicialização de variáveis sempre que for necessário;
 - dentre outros critérios.