

MC202 (Estruturas de Dados) - 1s2021

Tarefa 10: Filas de prioridade

Prof^o Ricardo Dahab

Assistente: Elisa Dell'Arriva

Assistente: Jônatas Trabuco Belotti

Instituto de Computação - UNICAMP

Sobre a Tarefa

O objetivo desta tarefa é que o(a) aluno(a) se familiarize com a criação e manipulação de filas de prioridade. O exercício consiste em realizar uma sequência de operações de inserção de elementos em um heap de máximo e de extração do elemento de valor máximo. Após cada operação, deve-se imprimir o heap resultante.

Filas de prioridades

Nesta tarefa, a *struct* que representa uma fila de prioridade deve, obrigatoriamente, ser a seguinte.

```
typedef struct fp {  
    int *valor, n, tamanho;  
} fp;
```

em que *n* é o número de elementos no heap, *tamanho* é o tamanho máximo da heap e *valor* é um vetor de elementos, que são números inteiros. O tamanho máximo do heap é 100.

Funções de manipulação do heap

A seguir, apresentamos as assinaturas das funções cuja implementação é obrigatória. Somente essas funções podem ser utilizadas para manipular o heap.

```
fp* cria_heap(int tamanho)  
  
void insere(fp *heap, int valor)  
  
int extrai_maximo(fp *heap)  
  
void escreve(fp *heap)
```

É permitida a criação de funções auxiliares, mas a manipulação do heap deve ser apenas por meio das funções descritas acima.

Formato da entrada e saída

A entrada é dada por uma sequência de operações, uma por linha. O primeiro caractere de cada linha é um identificador da operação a ser realizada.

I: indica inserção e é seguida do valor a ser inserido;

M: indica extração do valor máximo; e

F: indica o fim da sequência de operações.

Na operação de extração do máximo, deve-se considerar que a esquerda tem precedência em relação à direita, ou seja, ao reorganizar o heap, sempre consulte primeiro o filho à esquerda.

Após cada operação realizada, deve-se escrever o heap resultante, seguido de uma quebra de linha. Os valores devem estar separados por um espaço (o último valor também é seguido de um espaço). Caso o heap resultante seja vazio, deve-se imprimir "VAZIO", seguido de uma quebra de linha. A seguir, apresentamos três exemplos de entradas e suas respectivas saídas.

Exemplo 1.

Entrada:

```
I 13
I 55
I 51
M
F
```

Saída:

```
13
55 13
55 13 51
51 13
```

Exemplo 2.

Entrada:

```
I 8382
I 4234
M
I 6323
I 9746
I 3005
I 8146
I 5935
I 87
F
```

Saída:

```
8382
8382 4234
4234
6323 4234
9746 4234 6323
9746 4234 6323 3005
9746 8146 6323 3005 4234
9746 8146 6323 3005 4234 5935
9746 8146 6323 3005 4234 5935 87
```

Exemplo 3.

Entrada:

```
I 9872
I 796
I 1258
M
M
M
F
```

Saída:

```
9872
9872 796
9872 796 1258
1258 796
796
VAZIO
```

Importante

- É obrigatório utilizar a *struct* descrita neste enunciado para representar um heap. O não cumprimento dessa regras implicará em nota zero nesta tarefa.
- É obrigatório implementar e utilizar todas as assinaturas de funções fornecidas no enunciado. O não cumprimento dessa regras implicará em nota zero nesta tarefa.
- É obrigatório liberar toda memória alocada dinamicamente. Para fazer essa verificação, recomendamos o uso da ferramenta *valgrind*. Caso haja memória não liberada, serão descontados dois pontos da nota final desta tarefa.

Alguns avisos e lembretes

- A página da disciplina no SuSy é <https://susy.ic.unicamp.br:9999/mc202abc>.
- Para submeter, utilizem somente os dígitos numéricos do RA e a senha da DAC.
- Espere, no mínimo, 60 segundos entre uma submissão e outra no SuSy, para evitar problemas na entrega.
- O número máximo de submissões é 20.
- Esta tarefa tem peso 1 (um).
- O prazo de entrega desta tarefa é 05/07/2021, às 23h59. Serão aceitas submissões atrasadas até o dia 08/07/2021 e serão aplicados os descontos de nota estabelecidos conforme o número de dias em atraso.