

# MC202EF - Estruturas de Dados

## Lab 07

(O conteúdo necessário para realizar este laboratório vai até a unidade:  
Fila de Prioridade e Heap-Sort)

PED: Márcio de Carvalho Saraiva

PAD: Anderson, Mateus, Victor

Professor: Lehlilton Lelis Chaves Pedrosa

20 de outubro de 2016

## 1. Problema<sup>1</sup>

O termo cache pode ser entendido como uma área de armazenamento onde dados frequentemente usados são guardados para um acesso futuro mais rápido, poupando tempo e recursos. Um exemplo de uso de caches estão nos navegadores de internet. Não é por acaso que sites que você frequenta constantemente costumam abrir mais rápido do que os que visita pela primeira vez. Isso se deve ao fato do cache do navegador salvar a estrutura básica das páginas que você está habituado a visitar, poupando o tempo de download delas em exibições futuras. Na época da internet discada era muito fácil perceber quando uma página estava em cache.

No entanto, caches têm uma tendência desagradável para encher. Quando isso acontece alguns objetos devem ser retirados do cache para criar espaço para novos objetos. Chamamos de política de substituição o método usado para a escolha do objeto que será removido do cache. Você foi contratado por uma empresa que está desenvolvendo novas políticas de substituição e você irá ajudar a avaliar esses novos algoritmos.

Dados um tamanho de cache, os possíveis elementos e uma sequência de acessos aos elementos, você deve encontrar uma solução que minimize o número de vezes que os objetos são inseridos no cache. Os acessos são atendidos em sequência e para que um elemento seja acessado ele deve estar no cache, de modo que quando um acesso a um elemento  $x$  é feito,  $x$  deve ser inserido no cache se ele já não estiver lá.

Todos os objetos têm o mesmo tamanho e um objeto em cache é sempre válido. Quando o sistema é iniciado o cache está vazio.

---

<sup>1</sup> Adaptado originalmente por Márcio Felix.

## 2. Entrada

A primeira linha contém três inteiros  $c$ ,  $n$  e  $a$ , onde  $c$  é o tamanho do cache,  $0 < c \leq 10^4$ ,  $n$  é a quantidade de elementos existentes no sistema,  $0 < n \leq 10^5$ , e  $a$  é a quantidade de solicitações de acessos a elementos,  $0 < a \leq 10^5$ . As próximas  $a$  linhas contém um único inteiro entre 0 e  $n - 1$  indicando que elemento é acessado. A primeira linha corresponde ao primeiro objeto acessado e a última linha ao último. Exemplo de entrada:

```
2 3 5
0
1
2
2
1
```

## 3. Saída

A saída é um único inteiro que representa o número mínimo de vezes que os objetos precisam ser lidos no cache para atender a lista de acessos na entrada. Note que a solução está diretamente relacionada com a decisão de que elemento deve ser removido do cache quando precisar de espaço. Além disso, tendo toda a sequência de acessos aos elementos a priori é fácil tomar essa decisão: o elemento a ser retirado do cache tem que ser exatamente aquele que vai demorar mais a ser lido de novo. A saída do exemplo será:

```
3
```

Neste exemplo temos um cache de tamanho 2, o número de elementos é 3 e temos 5 solicitações de acessos. Uma solução ótima seria a seguinte:

cache inicialmente vazio

0 - cache: 0 - coloca 0 no cache (solucao++)

1 - cache: 0, 1 - coloca 1 no cache (solucao++)

2 - cache: 2, 1 - tira 0 do cache e coloca 2 (solucao++)

2 - cache: 2, 1 - 2 já está no cache

1 - cache: 2, 1 - 1 já está no cache.

Note que no primeiro acesso ao elemento 2 tiramos o 0 do cache. Se o 1 tivesse sido removido nossa solução seria pior.

## 4. Dicas

- Lembrem-se de liberar o espaço após o uso.
- Responda: é melhor retirar do cache um elemento que será necessário daqui a duas operações ou outros que só será necessário daqui a dez operações?

## 5. Avaliação

5.1 As notas desse laboratório será calculada da seguinte maneira:

- 7 pontos proporcionais ao número de casos de teste corretos;
- 3 pontos referentes à qualidade de código: legibilidade, algoritmo, memory leak, boas práticas de programação (comentários no código, escolha do nomes para variáveis, reutilização de funções que possam melhorar a apresentação do código, etc...).

5.2 Critérios avaliados neste laboratório:

Além de passar nos casos de teste, é obrigatório:

- Seu programa deve usar uma fila de prioridade
- 

## 6. Entrega

A submissão de código deve ser feita no Run.codes em no máximo 10 tentativas até o dia 4/11/16 às 23:59:59.