1ª Prova de EDA-II - Curso: Engenharia de Software

Data: 2 de Maio **Horário limite:** 20:00

(40 pontos) 1) Considere a implementação de uma tabela hash utilizando encadeamento separado, listas encadeadas, para tratamento de colisões. Responda as perguntas abaixo:

a) (20 pontos) Escreva uma função `grau` que recebe como parâmetro a tabela hash e imprima quantos *itens* colidem em cada hash. Qual é o custo aproximado desta função?

- É possível responder o `grau` de uma hash qualquer em tempo constante, i.e $\mathcal{O}(1)$?
- Que modificação na estrutura deve ser feita para poder responder esta pergunta? Se o custo passa a ser constante, qual é a troca (*tradeoff*) para isso, explique? (ex. Mais consumo de CPU, Menos uso de memória...)

b) (20 pontos) Qual o efeito de haver muitas colisões (aproximadamente N) em uma hash específica? Afeta, de alguma forma, a operação de inserção ou busca na tabela hash? Justifique e estime o custo das operações, caso sejam afetadas.

(25 pontos) 2) Escreva uma implementação eficiente da operação Remove com parâmetros `(A, n, i)`. Ela deve remover o nó `i` do max-heap `A[1 .. n]` e armazenar os elementos restantes, em forma de max-heap, no vetor `A[1 .. n-1]`.

(10 pontos) 3) Insira os itens `E D A D O I S T D O I S`, nessa ordem, em um heap decrescente inicialmente vazio. Mostre o heap que resulta em formato de Vetor e em formato de Árvore.

(25 pontos) 4) Dado um vetor de inteiros, sua tarefa é encontrar a k-ésima ocorrência (da esquerda para a direita) de um inteiro v no vetor. Para tornar o problema mais difícil (e mais interessante!), você deve responder a m consultas deste tipo.

Entrada

Há vários casos de teste. A primeira linha de cada caso de teste contém dois inteiros n e m ($1 \leq n, m \leq 100.000$), o número de elementos no vetor e o número de consultas a serem respondidas, respectivamente. A próxima linha contém n inteiros positivos não maiores que 1.000.000, que descrevem o vetor. As próximas m linhas contém dois inteiros k e v cada ($1 \leq k \leq n$, $1 \leq v \leq 1.000.000$), descrevendo as consultas.

O arquivo de entrada termina com fim-de-arquivo (EOF).

Lembre que a consulta deve ser eficiente!

Saída

Para cada consulta, imprima o índice do vetor (1-indexado) da ocorrência solicitada. Se tal ocorrência não existe, imprima 0 ao invés.

Entrada:	**Saída:**

`8 4`
`1 3 2 2 4 3 2 1`
`1 3`
`2 4`
`3 2`
`4 2`
	`2`
	`0`
	`7`
	`0`

Antes de implementar a solução deste problema é preciso refletir, responda as perguntas abaixo:

a) (5 pontos) Qual é o pior caso de entrada para este problema?

b) (10 pontos) Qual é o custo da consulta no pior caso? É possível fazer melhor que $\mathcal{O}(n)$? Explique a sua solução e diga o custo aproximado da operação de busca.

c) (10 pontos) Para poder realizar a busca melhor que $\mathcal{O}(n)$ explique como deverá ser armazenado o vetor de entrada e quanto custa a organização da entrada.