

Nome do aluno: _____ Data: 07/12/2018
 Resolva as questões abaixo identificando-as claramente na folha de respostas. Mantenha o silêncio na sala (mantendo desligado aparelhos eletrônicos). A interpretação das questões faz parte da prova. Horário limite: 15:30

(10 pontos) 1) Considere as técnicas de busca sequencial, busca binária e busca baseada em hashing:

- Descreva as vantagens e desvantagens de cada uma dessas técnicas, indicando em que situações você usaria cada uma delas.
- Qual é a eficiência de utilização da memória (relação entre o espaço necessário para dados e o espaço total necessário) para cada método?

(20 pontos) 2) Matriz esparsa

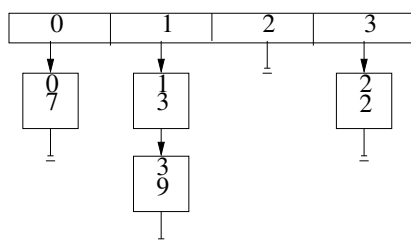
A estratégia clássica de representar uma matriz esparsa é com um vetor de colunas, e em cada índice deste vetor temos uma lista encadeada composta pelos elementos não nulos da coluna.

Observe na tabela 1 uma representação padrão de matriz, representada em um grid $M \times N$, no caso 4×4 . Na tabela 1 temos uma matriz esparsa, ou seja, a maior parte de seus elementos são nulos. Agora observe que a figura 1 representa esta matriz utilizando a estratégia clássica, comentada acima.

Tabela 1: Matriz Esparsa em uma representação padrão de Matriz

	0	1	2	3
0	7	0	0	0
1	0	3	0	0
2	0	0	0	2
3	0	9	0	0

Figura 1: Representação de Matriz esparsa com vetor de colunas e lista encadeada. Esta é a representação da matriz acima.



Considerando que uma matriz esparsa armazena números inteiros de 64bits e que os ponteiros também possuem 64bits e a representação clássica, responda:

- (5 pontos) Quando esta representação deixa de ser vantajosa quando comparada com uma representação padrão de matriz? Explique.
- (10 pontos) Escreva uma função **grau** que recebe como parâmetro o vetor de colunas da matriz e imprima quantos elementos não nulos existe em cada coluna. Qual é o custo aproximado desta função?
- (5 pontos) É possível responder o **grau** de uma única coluna em tempo constante, i.e $\mathcal{O}(1)$? Que modificação na estrutura deve ser feita para poder responder a esta pergunta? Se o custo passa a ser constante, qual é a troca (*tradeoff*) para isso?

(20 pontos) 3) Um animado aluno está criando um sistema centralizado que recebe submissões de problemas, no estilo do MOJ, porém ele recebe submissões de diversas regiões do país.

O interessante é que o sistema processa as submissões a cada 5 minutos e por isso recebe muitos dados. Cada região envia uma lista encadeada, ordenada pelo tempo decorrido da prova em que a submissão foi feita, ou seja, se uma submissão foi feita após 5 minutos de prova ela terá valor 5, se for 15 minutos, será 15.

O sistema deverá processar todas submissões, de todas as regiões. A ordem de processamento das submissões deverá ser ordenada pelo tempo decorrido da prova, ou seja, a submissão mais antiga (menor valor de tempo decorrido da prova), independente da região, é a que deverá ser feita primeiro.

O problema é relativamente simples, pois como todas as listas recebidas já estão ordenadas, basta criar um algoritmo que junte (intercale) todas as listas encadeadas em uma única lista, porém o seu algoritmo **NÃO** deve alocar novas células de memória.

Foi pedido a você que implemente 2 funções, uma chamada `intercala4` e outra `intercala8`, estas funções recebem, respectivamente, 4 e 8 ponteiros do tipo `struct no`, sendo, cada um, o início de uma lista encadeada ordenada pelo `tempo_de_submissao` e deverá retornar um único ponteiro do tipo `struct no` contendo uma lista resultante da intercalação das listas recebidas como parâmetro. Você deverá implementar da forma mais eficiente possível. Você receberá em sua função somente os endereços dos nós (sem a cabeça) das listas.

Dica: Você pode achar conveniente implementar a função `intercala2`. É possível criar *wrappers*?

A struct da lista encadeada é:

```
1 struct no {
2     int tempo_de_submissao;
3     struct codigo;
4     struct no *proximo;
5 };
```

A questão 4 está na próxima página.

(50 pontos) 4) Um aluno da UnB/Gama (Universidade de Brasília no GAMA) , está implementando um novo e revolucionário sistema de consulta de dados. Esse sistema será utilizado para fazer consultas no sistema *Rettiwt* onde as pessoas poderão colocar mensagens sobre variados temas e os usuários poderão fazer consultas sobre as mensagens.

O nosso nobre colega está tendo alguns problemas na implementação das consultas de mensagens. Para facilitar a abstração do problema foi no passado uma versão simplificada do problema, onde temos:

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:
	2
8 4	0
1 3 2 2 4 3 2 1	7
1 3	0
2 4	
3 2	
4 2	

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

- (5 pontos) Qual é o pior caso de entrada para este problema?
- (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.
- (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, é possível fazer melhor que $\mathcal{O}(2n + n^2 \log n)$?
- (25 pontos) Implemente uma solução eficiente para este problema e estime os custos de organização do vetor de entrada e da consulta no pior caso.

Esta página seria intencionalmente deixada em branco se não fosse por este aviso.