1800 30345 - ARTHUR SENA

2ª PROVA DE EDA-II - Curso: Engenharia de Software

Data: 21 de Março **Horário limite:** 9:55

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

a) Descreva as vantagens e desvantagens de cada uma dessas técnicas, indicando em que situações você usaria cada uma delas.

b) Qual é a eficiência de utilização da memória (relação entre o espaço necessário para dados e o espaço total necessário) para cada método?

(15 pontos) 2) Imagine que você tenha um *bug* em sua implementação de tabela hash utilizando hash-dupla (double-hashing) de tal forma que a primeira ou a segunda função de hash retornam sempre o mesmo valor (porém diferente de 0). Descreva o que ocorre (exemplo: os custos de inserção e busca permanecem o esperado?) quando:

a) a primeira hash está errada;

b) a segunda hash está errada;

c) ambas funções de hash estão erradas.

(15 pontos) 3) A respeito de Fila de Prioridades

a) Para que serve as operações de `fixup` e `fixdown`

b) A Fila de Prioridades é uma estrutura de dados e Busca? Justifique.

c) Implemente a função `void PQchange(struct pq *PQ,int k)`, onde

- `PQ` é a fila de prioridades implementada em vetor
- `k` é o índice do elemento no vetor `PQ` que teve a prioridade modificada

(15 pontos) 4) Degustação

A Empresa de Degustação Aguda (EDA) está criando uma bebida apurada e envelhecida nos melhores barris do mundo. E para determinar o melhor sabor, esta nobre empresa decidiu pedir para que você fizesse o processamento da string de escolhas.

A análise acontece da seguinte forma:

- Uma string com as letras das escolhas é passada para o seu programa;
- A posição em que cada uma começa é importante, a primeira começa na posição 0;
- Você precisa contar o tamanho das sequências formadas pelo mesmo caractere, por exemplo: `aabbbcaaaa`
  - As sequências do exemplo acima são:
  - *a* começando na posição 0 composta por 2 ocorrências;
  - *b* começando na posição 2 composta por 3 ocorrências;
  - *c* começando na posição 5 composta por 1 ocorrência;
  - *a* começando na posição 6 composta por 4 ocorrencias
    - veja que contabilizamos as sequências com os mesmos caracteres independetemente.

Implemente a função `void resolve(char *S)`, que recebe como argumento a string `S`, e imprime diversas linhas contendo:

- Cada linha deve conter três dados, são eles: um inteiro =I=; um caractere `C`, e; um inteiro `P`; representando respectivamente o tamanho da sequência; o caractere da sequência, e; a posição que o caractere começou na string `S` original.
- A saída deverá estar ordenada de maneira não crescente pelo indexador `I` e em caso de empate considere a sequência que apareceu antes na entrada.

Considere que as seguintes funções, e macros, estão a sua disposição:

```
less(A,B);lesseq(A,B);exch(A,B);cmpexch(A,B)
void quicksortM3(Item *v,int l,int r)
void mergesort(Item *v,int l,int r)
void insertionsort(Item *v,int l,int r)
void selectionsort(Item *v,int l,int r)
int separa(Item *v,int l,int r)
```

Exemplo, para a entrada aabbbcaaaa a saída deverá ser:

```
4 a 6
3 b 2
2 a 0
1 c 5
```

(45 pontos) 5) 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 mensageus.

O nosso nobre colega está tendo alguns problemas na implementação das consultas de mensagens. Para facilitar a abstração do problema foi nos 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:

```
8 4
1 3 2 2 4 3 2 1
1 3
2 4
3 2
4 2
```

Saída:

```
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.

d) (20 pontos) Implemente uma solução eficiente para este problema e estime os custos de organização do vetor de entrada e da consulta no pior caso.