Aluno(a):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Matrícula:\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| 1a) | 1.0 |  |
| 2a) | 3.0 |  |
| 3a) | 3.0 |  |
| 4a) | 2.0 |  |
| 5a) | 1.0 |  |
|  | 10.0 |  |

1. A prova é individual e sem consulta.
2. A interpretação faz parte da questão. Em caso de dúvida escreva a dúvida e a sua interpretação na resposta.
3. O tempo de prova é 1:45 h.
4. As respostas devem seguir as questões. Caso precise de rascunho use o verso da folha.
5. A prova pode ser feita a lápis.

**Questão 1** (1.0 pontos):

a) (0,5 ponto) Seja *T* uma tabela de dispersão (*hash table*) implementada com *encadeamento externo*. Assuma que *T* possui tamanho *n* e que a função de dispersão seja dada por:

Assuma que o total de elementos a serem inseridos seja de aproximadamente 600. Qual valor de *n* você escolheria? Explique sua resposta.

b) (0,5 ponto) Seja *T* uma tabela de dispersão (*hash table*) implementada com *encadeamento interior*. Assuma que *T* possui tamanho *n*. Defina uma função de dispersão adequada e simples (não é necessário escrever a função em C; basta defini-la matematicamente). Explique sua resposta.

Resp:

**Questão 2** (3.0 pontos): Escreva um rotina em C que receba como entrada um *max heap* e uma percentagem *p*, modifique a prioridade *x* do maior elemento do heap para *p.x* e reorganize a estrutura para que mantenha as propriedades de um *max heap*. A rotina deve retornar 1 se a alteração ocorreu corretamente e 0 em caso contrário.

A sua implementação deve fazer o menor número possível de comparações e movimentações (e não deve usar as operações sobre heaps pré-definidas).

Assuma que o *max heap* seja definida como abaixo:

struct \_heap {

int max; /\* tamanho maximo do heap \*/

int pos; /\* proxima posicao disponivel no vetor \*/

float\* prioridade; /\* vetor das prioridades \*/

};

typedef struct \_heap Heap;

Assuma ainda que a assinatura da rotina seja:

int heap\_altera(Heap\* heap, float p)

Resp:

**Questão 3** (3.0 pontos): Considere o problema de representar o grupo sanguíneo de um conjunto de pessoas, que pode variar ao longo do tempo, de tal forma que seja rápido determinar se duas pessoas possuem o mesmo grupo sanguíneo.

a) (1,0 ponto) Qual estrutura de dados, dentre as apresentadas em sala, seria mais eficiente para resolver o problema acima? Explique sua resposta.

c) (2,0 pontos) Defina em C a estrutura de dados escolhida no item (a) e implemente uma rotina para determinar se duas pessoas possuem o mesmo grupo sanguíneo. A rotina não deverá chamar as operações definidas em sala para a estrutura escolhida.

Resp.

**Questão 4** (2.0 pontos): Considere o problema de representar 2 subconjuntos de um grupo *G* de pessoas, definidos como:

*G1* = { *x* ∈ *G* / *x* tem 21 anos ou mais }

*G2* = { *x* ∈ *G* / *x* é do sexo feminino }

a) (0,5 ponto) Qual estrutura de dados, dentre as apresentadas em sala, seria mais eficiente para representar os 2 subconjuntos acima e implementar operações sobre eles? Explique sua resposta.

c) (1,5 pontos) Defina em C a estrutura de dados escolhida no item (a) e implemente uma rotina que receba como entrada *G1* e *G2* e devolva o subconjunto *H* de *G* definido como:

*H* = { *x* ∈ *G* / *x* tem 21 anos ou mais e *x* não é do sexo feminino }

A rotina não deverá chamar as operações definidas em sala para a estrutura escolhida.

Resp:

**Questão 5** (1.0 ponto):

a) (0,5 ponto) Considere a tabela de dispersão estendida, *T*, ***parcialmente*** representada abaixo. Quantos apontadores em *T* apontam para *B1* ? Caso haja mais de um apontador, em quais posições da tabela *T* eles estarão armazenados? Explique sua resposta.

*T*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *i =3* |  |  |  |
| 0 |  |  |  | *B1* |
| 1 |  |  |  | *j1 =2* | |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |

b) (0,5 ponto) Considere a tabela de dispersão estendida, *T*, representada na figura abaixo. Suponha que cada bloco contenha 2 chaves, como ilustrado na figura abaixo. Mostre como a estrutura ficará após a remoção de *K31* e *K32*, assumindo que a estrutura deve economizar espaço sempre que possível. Explique sua resposta.

*T*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *i =2* |  |  |  |
| 0 |  |  | *K11* | *B1* |
| 1 |  |  | *K12* | *j1 =2* | |
| 2 |  |  |  |  |
| 3 |  |  | *K21* | *B2* |
|  |  |  | *K22* | *j2 =2* | |
|  |  |  |  |  |
|  |  |  | *K31* | *B3* |
|  |  |  | *K32* | *j3 =2* | | |
|  |  |  |  |  |
|  |  |  | *K41* | *B4* |
|  |  |  | *K42* | *j4 =2* | |

Resp.: