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

|  |  |  |
| --- | --- | --- |
| 1a) | 2,5 |  |
| 2a) | 2,5 |  |
| 3a) | 2,5 |  |
| 4a) | 2,5 |  |
|  | 10.0 |  |

**LEIA COM CUIDADO**

1. A prova é individual e sem consulta.
   1. **Qualquer tentativa de “cola” resultará na anulação da prova do aluno ou dos alunos envolvidos**.
   2. Os aparelhos celulares deverão permanecer desligados e guardados fora do alcance durante toda a prova. **Aparelhos celulares ligados ou de alguma forma visíveis serão tratados como tentativa de “cola”**.
2. A interpretação faz parte da questão.
   1. **Não há perguntas durante a prova.**
   2. Em caso de dúvida escreva a dúvida e a sua interpretação na resposta.
3. O tempo de prova é 1:45 h.
4. **Após o início da prova, não será possível sair e voltar à sala.**
5. As respostas devem seguir as questões. Caso precise de rascunho use o verso da folha.
6. A prova pode ser feita a lápis.

**Questão 1 (2,5 pontos)** Em cada um dos casos abaixo indique qual ou quais rotações devem ser feitas para rebalancear a árvore AVL após a operação indicada. Explique a sua resposta, indicando os fatores de balanceamento dos nós que provocam as rotações e a forma da árvore a cada rotação **(sem esta explicação a questão não será corrigida)**.

(a) (1,0 ponto) Operação: **Inserção da chave 6**.

5

3

10

12

7

(b) (1,0 ponto) Operação: **Remoção da chave 7, trocando-se 7 com a sua antecessora.**

11

5

2

7

1

3

10

6

9

(c) (0,5 ponto) Suponha que imediatamente após a inserção da chave 6 no item (a), seja inserida a chave 4. Baseando-se neste cenário, explique se, e como, seria possível otimizar o processo de inserção.

**Resposta**

**Questão 2 (2,5 pontos)** Considere a árvore B de ordem 5 abaixo:

100

53 77 123 200

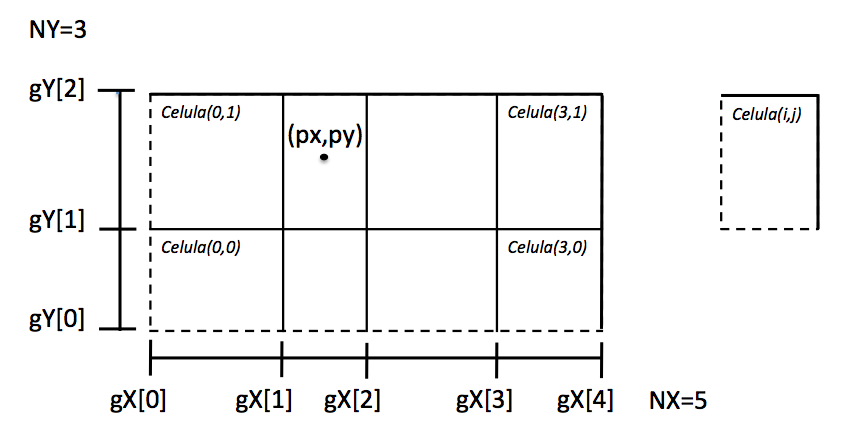
10 40 60 70 80 90 110 113 130 138 140 170 230 236 242 245

Realize as seguintes operações, indicando as transformações (divisão, redistribuição ou concatenação) executadas e a forma da árvore após cada transformação:

1. (0,5 ponto) Inserção de 243 na árvore original.
2. (1,0 ponto) Remoção de 123 da árvore original, trocando-se 123 pela sua antecessora.
3. (1,0 ponto) A troca de 123 pela sua antecessora, exigida no item (b), seria a melhor forma de remover 123? Explique a sua resposta exibindo a alternativa escolhida e comparando-a com a alternativa do item (b).

**Resposta**

**Questão 3 (2,5 pontos)** Considere uma grade não regular bidimensional tal que (ver figura abaixo): as bordas lateral esquerda e inferior não pertencem à grade; para cada célula, as bordas lateral esquerda e inferior não pertencem à célula.



Cada célula da grade armazena uma lista encadeada de pontos.

Considere os seguintes tipos que representam a grade:

typedef struct lista Lista;

struct lista {

float x, y; /\* ponto na lista \*/

Lista\* prox; /\* ponteiro para próximo elemento da lista \*/

};

typedef struct grade Grade;

#define NX 200 /\* número de elementos do vetor gX \*/

#define NY 135 /\* número de elementos do vetor gY \*/

struct grade {

float gX[NX]; /\* vetor que define as divisões no eixo X da grade \*/

float gY[NY]; /\* vetor que define as divisões no eixo Y da grade \*/

Lista\* prim[NX][NY]; /\* lista por célula (inicializada com NULL) \*/

};

Implemente uma função que receba como entrada uma grade *g*, com a definição acima, e um ponto *(px, py)*, como na figura, e pesquise se o ponto ocorre na grade. A função deve seguir o seguinte protótipo:

int pesq(Grade\* g, float px, float py);

e deve retornar:

0 se o ponto não existir na grade, ou estiver fora da grade

1 se o ponto existir na grade

**Resposta**

**Questão 4 (2,5 pontos)** Considere uma tabela de dispersão estendida, parcialmente representada na figura abaixo, onde

i = número de bits de relevância

h = função de hash

H = tabela de hash, armazenando apenas ponteiros para os buckets

B = um bucket arbitrário

jB = informação adicional sobre B

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

i

H

|  |
| --- |
|  |
|  |
|  |

jB

B

* 1. (0,5 ponto) Seja k = h(K) o valor da função de hash para uma chave K. Escreva uma expressão em C que mapeia k e i em um valor x tal que H[x] aponta para o bucket onde K deve ser armazenada.
  2. (0,5 ponto) Suponha que o bucket B esteja completo e que existam exatamente 2 entradas de H, H[x] e H[y], apontando para B e que x < y. Ao inserir uma nova chave K em B, haverá overflow de B e um novo bucket B’ será criado; H[x] continuará apontando para B e H[y] apontará para B’.

Sob que condições a criação de B’ não resolverá o problema do overflow do bucket B? Explique sua resposta.

* 1. (0,5 ponto) Quando a situação em (b) ocorre, explique como a estrutura deve ser atualizada para resolver o problema.
  2. (1,0 ponto) Suponha que N seja o número máximo de bits de relevância que podem ser considerados. Explique qual novo problema poderá ocorrer no item (c) e como deve ser resolvido.

**Resposta**