Simulado da segunda avaliação

Algoritmos e Estrutura de Dados II (AE23CP)

Prof. Jefferson T. Oliva

Questionário

1. Dada a estrutura de dados abaixo para árvore B, implemente uma função que receba uma árvore B e retorne a quantidade itens pares de uma árvore B. Também, comente cada linha de código descrevendo o que ela faz.

```
typedef struct NodeB NodeB;

struct NodeB{
   int nro_chaves;
   int chaves[N - 1];
   NodeB *filhos[N];
   int eh_no_folha;
};
```

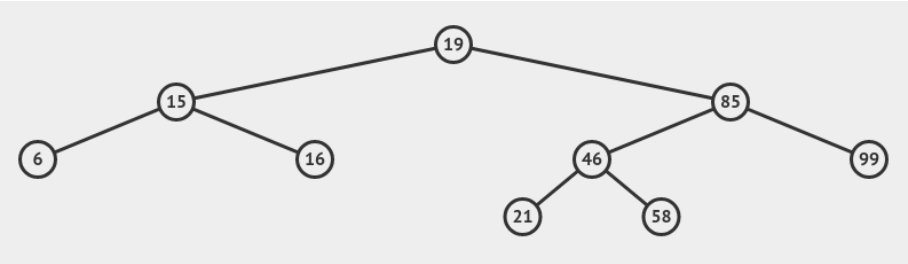
Resolução:

```
static void conta_par(NodeB *tree, int* contador){
   int i;
   if (tree != NULL){
       for (i = 0; i < tree->nro_chaves; i++){
           if (tree->chaves[i] % 2 == 0)
               (*contador)++;
           if (!tree->eh_no_folha)
               contar_par(tree->filhos[i], contador);
       }
       if (!tree->eh_no_folha) //após o laço acima, i será a posição da última página folha
           contar_par(tree->filhos[i], contador);
   }
}


int npar(NodeB *tree){
   int contador = 0;
   conta_par(tree, &contador);
   return contador;
}
```


Continua na próxima página...

2. Dada a árvore AVL abaixo:



Faça:

a) - Informe o fator de balanceamento para cada nó.

Resolução:

- 19: -1
- 15: 0
- 85: +1
- 6: 0
- 16: 0
- 46: 0
- 99: 0
- 21: 0
- 58: 0

b) - Mostre o passo-a-passo para a inclusão sequencial dos seguintes itens na árvore: 5, 4, 70, 22, 17. Também, se for o caso, indique as rotações executadas. Caso deseje desenhar o passo a passo da construção das árvores em papel, neste exercício é permitida a submissão de imagem.

Resolução:

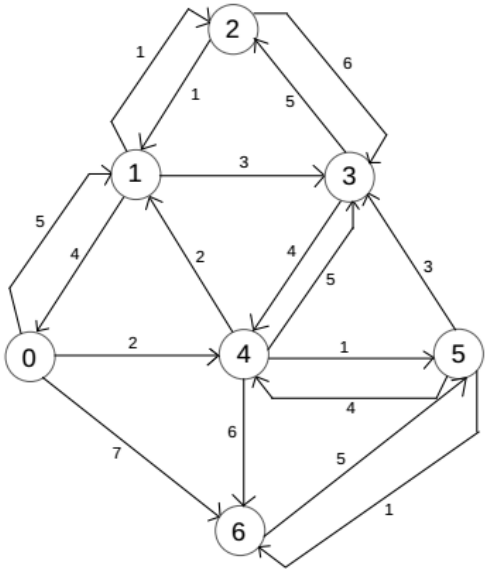
- 5: inserção ao lado esquerdo de 6; nenhuma rotação será necessária
- 4: inserção ao lado direito de 5; o nó 6 ficará desbalanceado. como o fator de balanceamento do nó 6 é positivo e de 5 é negativo, então a rotação é do tipo esquerda-direita; como resultado, o nó 5 passa ser filho esquerdo de 15 e os nós 4 e 6 passam a ser filhos esquerdo e direito, respectivamente, de 5.
- 70: Inserção ao lado direito de 58; consequentemente, o nó 85 fica desbalanceado com o fator de balanceamento positivo e a sua subárvore à esquerda (onde está o desbalanceamento) tem fator de balanceamento negativo, indicando que a rotação é do tipo esquerda-direita; então o nó 58 passa a ser filho direito de 19; 46 passa a ser filho esquerdo de 58; 85 passa a ser filho direito de 58 e 70 passa a ser filho esquerdo de 85; dessa forma, o nó 46 não tem mais descendente do lado direito.
- 22: inserido ao lado direito de 21, causando desbalanceamento no nó 46 (fator de balanceamento positivo); como a subárvore à esquerda de 46 (onde está o desbalanceamento) tem fator de balanceamento negativo, então a rotação é do tipo esquerda-direita será necessária; dessa forma, o nó 22 passa a ser filho direito de 58; os nós 21 e 46 passam a ser filhos esquerdo e direito, respectivamente, de 22.
- 17: inserido direito de 16, mas sem necessidade de rotação da árvore.
- Simulador de árvore AVL: `https://www.inf.ufsc.br/~aldo.vw/estruturas/simulador/AVL.html`

 Continua na próxima página...

3. (15 pontos) Dada uma árvore rubro-negra vazia. Mostre o passo-a-passo para a inclusão sequencial os seguintes itens: 13, 8, 16, 7, 4, 30, 45, 34, 1, 3. Também, se for o caso, indique as rotações executadas.

   Resolução: faça o teste no simulador de árvore rubro-negra: `https://www.inf.ufsc.br/~aldo.vw/estruturas/simulador/RB.html`

4. Dado grafo ponderado direcionado abaixo:



   Resolva as questões abaixo:

   a) - Qual o tamanho do grafo acima? Mostre como você chegou a tal número.
   Resolução: 7 vértices + 18 arestas = 25. Então o tamanho do grafo é 25.

   b) - O grafo é conexo? Justifique a sua resposta.
   Resolução: sim. Para todos os vértices há pelo menos uma aresta. Também, não há componentes isolados.

   c) - O grafo é fortemente conexo? Justifique a sua resposta.
   Resolução: Sim. Como o grafo é Hamiltoniano, todos os pares de vértices são alcançáveis.

Continua na próxima página...

d) - Demonstre a aplicação, passo a passo, do algoritmo de Dijkstra no grafo, iniciando do vértice 1. Como resultado, para cada vértice v apresente a menor distância para alcançá-lo (chave[v]) e o seu respectivo pai (π[v]). Também, apresente o comportamento da fila de prioridades.

Gabarito:

vértice	k	pi
0	4	1
1	0	-
2	1	1
3	3	1
4	6	0
5	7	4
6	8	5

Ordem de remoção de vértices da fila de prioridades: 1, 2, 3, 0, 4, 5, 6

e) - Transforme o grafo orientado acima em grafo simples (por exemplo, as arestas (1,2) e (2,1) se tornarão uma) não orientado e desconsidere os pesos das arestas. Em seguida, aplique, passo-a-passo, o algoritmo de busca em largura, considerando o vértice 1 como fonte. Como resultado, para cada vértice v, apresente a distância (d[v]) e o seu respectivo pai (π[v]). Também, apresente o comportamento da fila e mostre como ficou a árvore de busca em largura.

Gabarito:

vértice	k	pi
0	1	1
1	0	-
2	1	1
3	1	1
4	2	0
5	2	3
6	2	0

Ordem em que os elementos foram inseridos na fina: 1, 0, 2, 3, 4, 6, 5

Final do questionário.