MAC 323 – Estruturas de Dados

BCC – Segunda Prova – 16 de junho de 1998

NOME DO ALUNO :
ASSINATURA:

INSTRUÇÕES

1. A prova pode ser feita a lápis.
2. A prova consta de 6 questões. Verifique antes de começar a resolver a prova se o seu caderno de questões está completo.

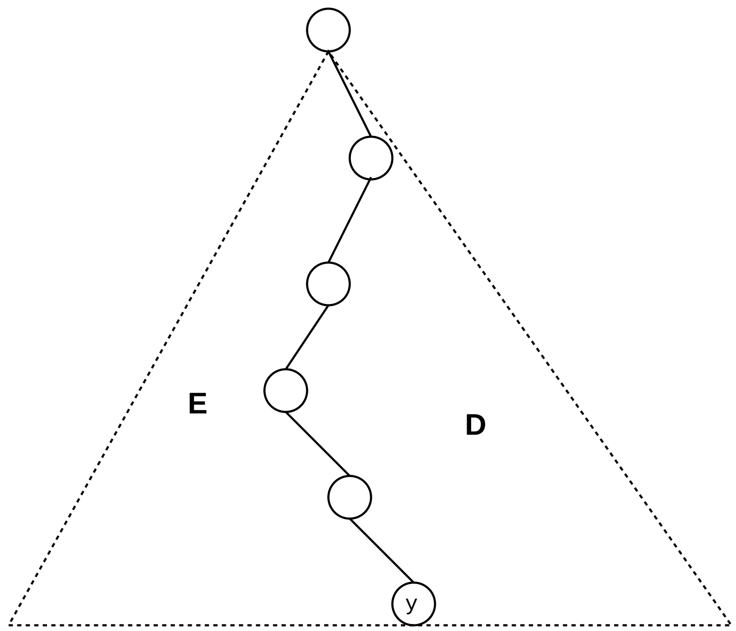
DURAÇÃO DA PROVA: 3 horas

BOA SORTE!!!!

	Nota
Questão 1	
Questão 2	
Questão 3	
Questão 4	
Questão 5	
Questão 6	
TOTAL	

BOLÃO: Brasil __ x __ Marrocos

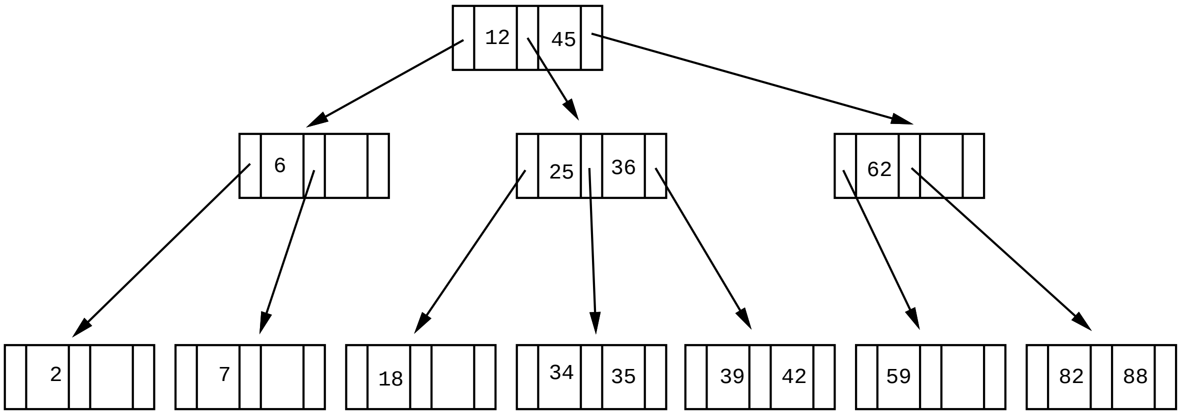
1. (valor 1.0+1.0 pontos)
Prove ou dê um contra-exemplo:
a. Em uma árvore de busca binária, seja C o conjunto de vértices visitados no caminho ao procurar um elemento y. Defina, E como sendo o conjunto de vértices à esquerda deste caminho e D os vértices à direita deste caminho (veja a figura abaixo). Então, para todo $x \in C$, $d \in D$ e $e \in E$, $d > x$ e $e < x$.



b. Uma árvore rubro-negra com n vértices tem altura $O(\log n)$.

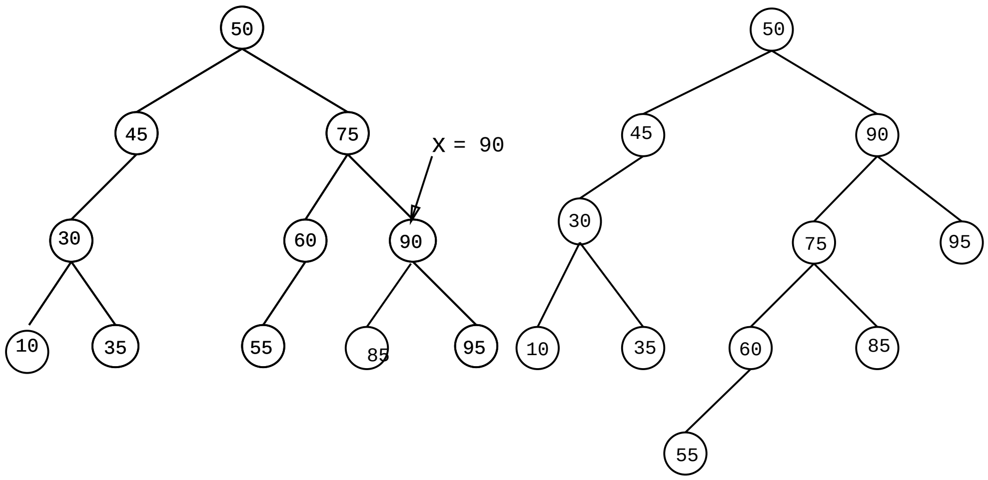
2. (valor 1.0+1.0 pontos)
   a. Diga o que é uma **árvore de busca binária ótima**. Em que sentido a árvore é ótima? Uma árvore ótima pode ser desbalanceada (altura $n-1$)? Se puder, dê um exemplo em que isto acontece.
   b. O que é a estrutura **union-find**? Dê uma aplicação para esta estrutura. Quais as operações que podem ser realizadas eficientemente com tal estrutura? Qual a complexidade de cada operação? Justifique.

3. (valor 1.0+1.0 pontos)
Considere a B-árvore da figura abaixo.



a. Simule **detalhadamente** o que ocorre ao remover o 7.
b. Simule **detalhadamente** o que ocorre ao inserir o 29.
Obs.: Se você não usar o algoritmo dado em sala de aula, explique que algoritmo de inserção/remoção você está utilizando.

4. (valor 1.0+1.0 pontos)
Dada uma árvore binária T, discuta a implementação mais eficiente possível e faça um algoritmo para os métodos abaixo. Se você achar conveniente, sugira uma implementação diferente da vista em sala de aula.
a. `Celula AncestralComumMaisPróximo(Celula` u`, Celula` v`)` que acha o primeiro ancestral comum de u e v
b. `void Sobe(Elemento` x`)` que recebe um elemento x e, se ele estiver na árvore e não for a raiz, sobe o nó contendo x de um nível, mantendo a estrutura de árvore de busca binária. Veja o exemplo a seguir.



5. (valor 2.0 pontos)
Dizemos que um grafo $G = (V, E)$ é **bipartido**, se o conjunto de vértices pode ser particionado em duas partes $V = A \cup B$, com $A \cap B = \emptyset$ tais que toda aresta do grafo tem exatamente uma ponta em A e uma ponta em B. Faça um método `boolean EBipartido()` que verifica se o grafo é ou não bipartido.

6. (valor 0.25+0.25+0.5 pontos)

   a. Faça uma crítica à disciplina: conteúdo, trabalhos, listas, EPs, professor, etc.

   b. Você lembra de algo que possa ser elogiado? Use o verso se necessário ☺

   c. Neste semestre eu tentei utilizar os recursos da rede como apoio para a disciplina. O que você achou disso? Quantas vezes você visitou a página da disciplina para ver algo ou tirar uma dúvida? Quantas vezes suas dúvidas foram tiradas por e-mails da mailing list?