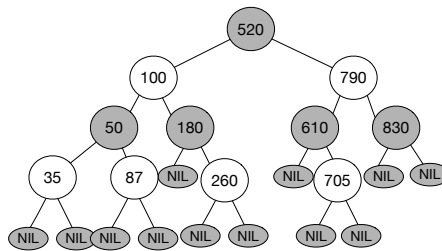


PUC-Rio – Estruturas de Dados – INF1010
Prova 2 – Turma 3wb – 18/10/2017

Responda as questões da prova nas folhas em separado distribuídas junto com a prova. As respostas podem ser escritas a lápis ou caneta. Indique claramente a questão que está respondendo e, quando cabível, os passos que seguiu para chegar a sua resposta.

Por favor guarde seu celular/tablet/o_que_for antes de começar a prova e não o utilize até sair da sala.

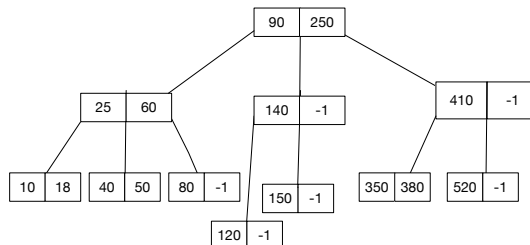
1. (1,5 ponto) Suponha a árvore rubro-negra mostrada a seguir, onde nós em cinza representam os nós negros e nós em brancos representam os nós vermelhos.



Como ficará a árvore se realizarmos cada uma das operações a seguir, *segundo o algoritmo de inserção para árvores rubro-negras visto na disciplina*. Cada uma das operações deve ser realizada sobre a árvore original. Desenhe (na folha a parte) a nova árvore para cada caso.

- (a) inserção com chave 730
- (b) inserção com chave 40

2. (2 pontos) Suponha a árvore B mostrada a seguir.



- (a) Como ficará a árvore se realizarmos cada uma das operações a seguir, *segundo o algoritmo de inserção para árvores B visto na disciplina* para inserção um a um? Cada uma das operações deve ser realizada sobre a árvore original. Desenhe (na folha a parte) a nova árvore para cada caso.
 - i. inserção com chave 360
 - ii. inserção com chave 48
 - iii. remoção com chave 80
 - iv. remoção com chave 140

3. (1 ponto) Considere a representação para árvores 2-3 a seguir, usada no laboratório.

```
typedef struct smapa Mapa;
struct smapa {
    int kp, kg; /* chaves: kp<kg, se kg existir. Se kg=-1, significa que ele não existe. */
    Mapa *pai;
    Mapa *esq;
    Mapa *meio;
    Mapa *dir;
};
```

Escreva uma função C que determine o número de elementos presentes em uma árvore.

```
int tamanho (Mapa* m);
```

4. (1,5 pontos) Considere a representação para heaps de prioridades a seguir, usada no laboratório, e a função `listap_cria`, que cria um novo *max-heap* com tamanho máximo `max`, recebendo uma lista de inteiros a serem usados para construir o estado inicial do *heap*.

```
typedef struct _listaprio ListaP;
struct _listaprio {
    int max; /* tamanho maximo do heap */
    int pos; /* proxima posicao disponivel no vetor */
    int* prioridade; /* vetor das prioridades */
};
ListaP* listap_cria(int max, int qtos, int* iniciais);
```

Escreva uma função C que receba uma lista de números inteiros e seu tamanho e retorne um novo array de inteiros contendo os números fornecidos em ordem crescente. Sua função deve utilizar a função `listap_cria` (supondo que esta já está implementada) e implementar o algoritmo de *heapsort* visto em sala.

```
int *cria_listaordenada (int *numeros, int qtos);
```

5. (2,5 pontos) Considere a representação abaixo para nós de uma árvore vermelho-negra:

```
typedef struct smapa Mapa;
struct smapa {
    int chave;
    int dado;
    char vermelho;
    struct smapa* esq;
    struct smapa *dir;
};
```

Uma árvore vermelho-negra é uma árvore binária que deve satisfazer as seguintes propriedades:

- (a) a raiz deve ser negra
- (b) um nó vermelho só pode ter filhos negros
- (c) todos os caminhos da raiz até uma folha devem conter o mesmo número de nós negros

Escreva funções em C para determinar se a segunda e terceira propriedade, respectivamente, valem em determinada árvore:

```
int filhosOK (Mapa *m);
/* Retorna 0 se existe algum nó vermelho com pelo menos um filho vermelho na árvore */
/* com raiz em m, 1 em caso contrário. */
int caminhosOK (Mapa *m);
/* Retorna 0 se existem caminhos com diferentes números de nós negros de m até duas */
/* folhas da árvore com raiz em m, 1 caso contrário. */
```

6. (0,5 ponto) Uma forma de tratar a colisão em tabelas de dispersão é por *sondagem linear*: caso a posição indicada pela função de dispersão já esteja cheia, a inserção procura a primeira posição vazia a partir desta, voltando para a posição 0 caso chegue ao final da tabela de dispersão. A tabela é um array de elementos onde cada um dos elementos contém a chave e um ponteiro para dados, e possivelmente outros campos. Como funciona a busca por um elemento com determinada chave quando se usa essa técnica?
7. (1,0 ponto) Suponha que estamos implementando uma lista de prioridades usando um *max-heap* implementado com array, como visto em sala. Se o *heap* for inicialmente criado vazio, e em seguida fizermos a sequência de inserções 300-150-400-90-179-450-340, como ficará o array? Preencha abaixo cada passo da inserção.

