MARCOS TIBBALL

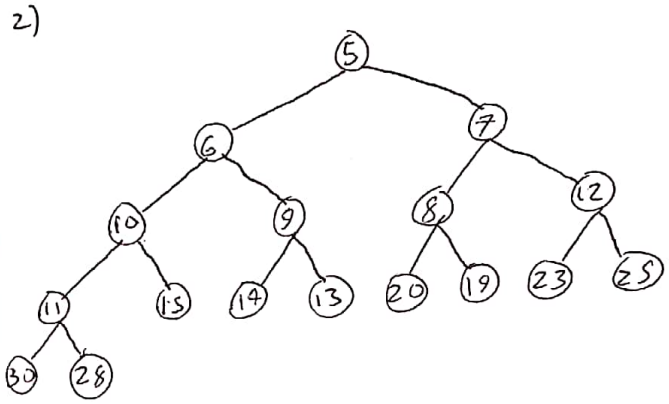
00302962

1)

LENDO EM PRÉ-ORDEM TEMOS:

33 - 15 - 41 - 38 - 34 - 47 - 43 - 49.

2)

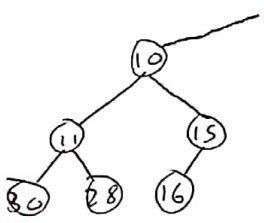


2) a)

SIM, ELA É UM HEAP BINÁRIO MÍNIMO COMPLETO, ONDE O VALOR DE CADA NODO É MENOR QUE O VALOR DE SEUS FILHOS.

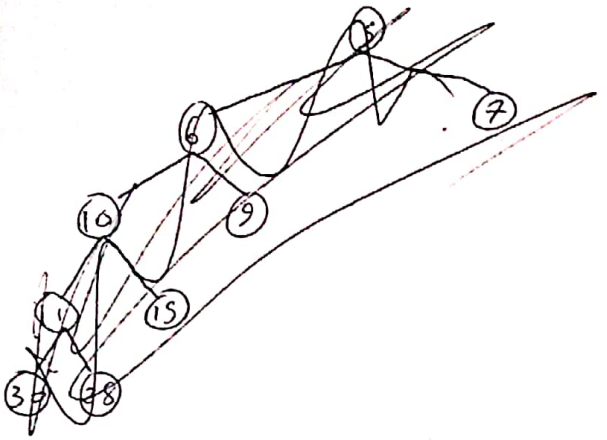
2) b)

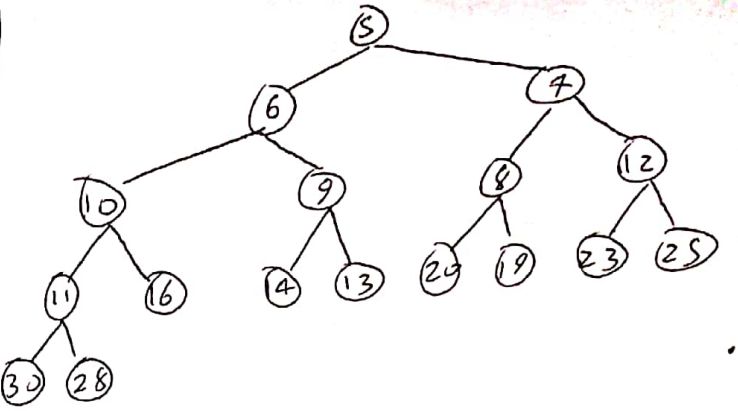
PARA ADICIONAR O ELEMENTO DE VALOR 16, O INSERIMOS NO ESPAÇO A ESQUERDA DO NÍVEL MAIS INFERIOR DO HEAP:

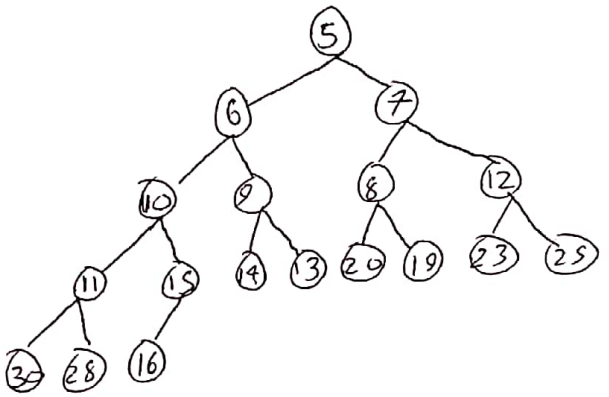


)

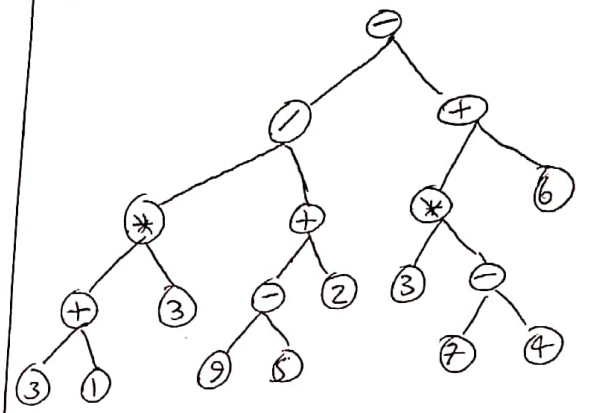
ENTÃO COMPARAMOS O ELEMENTO COM SEU PAI, COMO ELES ESTÃO NA ORDEM CERTA (FILHO > PAI), PARAMOS, TENDO:





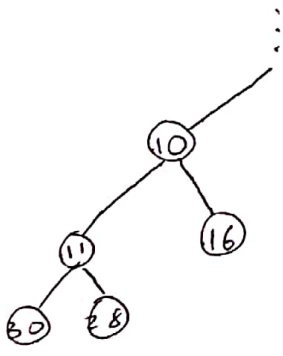


3) a)



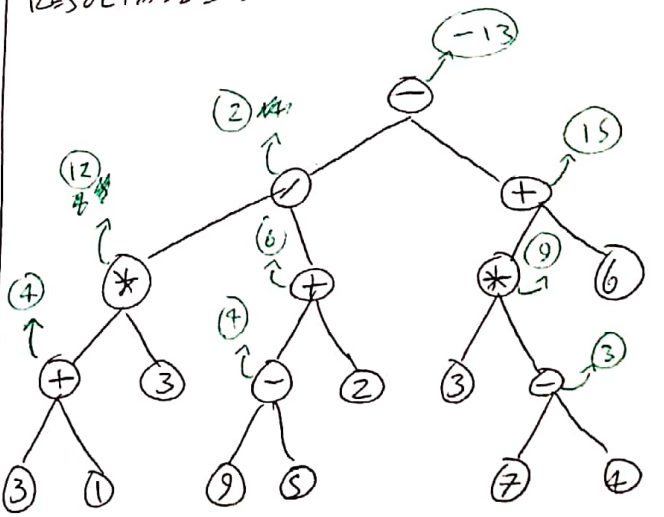
2) c)

PARA DEVOLVER UM ELEMENTO INICIALMENTE REMOVEMOS ELEMENTO DO HEAP E COLOCAMOS O ELEMENTO ~~DO~~ DO ÚLTIMO NÍVEL QUE MAIS ESTÁ NA DIREITA EM SEU LUGAR:



3) b)

CIRCULADO EM VERDE ENCONTRAM-SE OS RESULTADOS:



E ENTÃO CHECAMOS SE O NOVO NODO SE ADEQUA À REGRA DO HEAP, QUE É O CASO. ASSIM TEMOS:

4)

SIM, A LISTA DE ADJACÊNCIA É UMA LISTA ENCADEADA ~~QUE~~ ASSOCIADA COM CADA VÉRTICE DO GRAFO, ELA PERMITE FÁCIL ACESSO ÀS VÉRTICES ADJACENTES A CADA OUTRA VÉRTICE.

5)

CONSIDERANDO UMA DAS PROPRIEDADES BASE DA ÁRVORE BINÁRIA DE PESQUISA, O CAMINHAMENTO CENTRAL À ESQUERDA IRÁ SEMPRE OBTER AS INFORMAÇÕES DE FORMA DECRESCENTE. ASSIM:

```
VOID PERCORRER_EMORDEM (NODO * RAIZ) {

    IF (RAIZ == NULL) RETURN;

    // PERCORRE SUB-ÁRVORE ESQUERDA
    PERCORRER_EMORDEM (RAIZ -> ESQ);

    // PRINTA
    PRINTF ("%D", RAIZ -> INFO);

    // PERCORRE SUB-ÁRVORE DIREITA
    PERCORRER_EMORDEM (RAIZ -> DIR);

}
```

6)

NO ALGORITO DE DJIKSTRA FAZEMOS OS SEGUINTES PASSOS:

(I) MARCA TODOS OS NODOS COMO NÃO VISITADOS E CRIA UM CONJUNTO COM TODOS OS NODOS NÃO VISITADOS;

(II) ATRIBUIR PARA CADA NODO UM VALOR DE DISTÂNCIA DE INFINITO, EXCETO PARA O NODO INICIAL, QUE É O ATUAL;

(III) CALCULAR AS DISTÂNCIAS DO NODO ATUAL PARA CADA NODO VIZINHO, COMPARANDO AS NOVAS DISTÂNCIAS COM O VALOR ATUAL E ATRIBUIR O MENOR;

(IV) MARCAR O NODO ATUAL COMO VISITADO E REMOVER DO CONJUNTO DOS NÃO VISITADOS;

(V) SE O NODO DE DESTINO JÁ FOI VISITADO OU SE A MENOR DISTÂNCIA ENTRE OUTROS NODOS NÃO VISITADOS É ∞, PARAR;

(VI) SENÃO, SELECIONAR ~~ATRIBUIR~~ O NODO NÃO VISITADO COM MENOR DISTÂNCIA COMO O NODO ATUAL.

O ALGORITMO É CONSIDERADO GULOSO POR FAZER A ~~ESCOLHA LOCAL~~ MELHOR ESCOLHA LOCAL EM CADA ITERAÇÃO. ISSO É FEITO PELO ALGORITMO DE DJIKSTRA AO SELECIONAR SEMPRE A MENOR DISTÂNCIA LOCAL.