

Lista de exercícios 03

1-1) $h = 1$. $h = 5$.

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

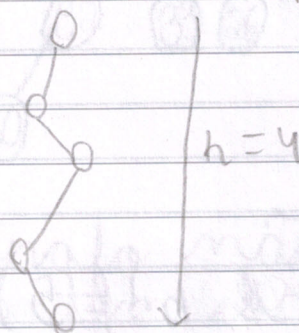
$2^4 = 16 - 31$

$2^5 = 32$
63

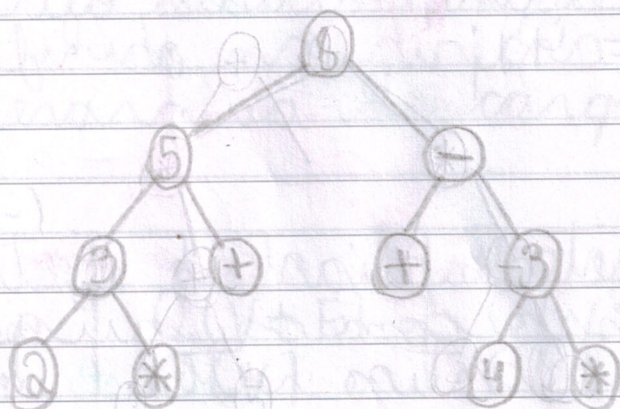
$O(\log n)$

$2^{h+1} - 1 = 63^{+1} - 1 = 63$

2-1)



3-1)



```
void imprimir2 (Arvore * r) {
```

```
    imprimir(r->esq);
```

```
    imprimir(r->dir);
```

```
    printf("%c", r->inf);
```

```
}
```


4-)

0	1	2	3	4

```
int hash (int elem) {
    return (elem % 5);
}

void hash_insert (int vet [], int num) {
    int h = hash (num);
    if (vet[h] == NULL)
        vet[h] = num;
    else {
        do {
            h = (h+1) % 5;
            if (vet[h] == NULL && h != 5)
                vet[h] = num;
        } while (h == 4);
    }
}
```

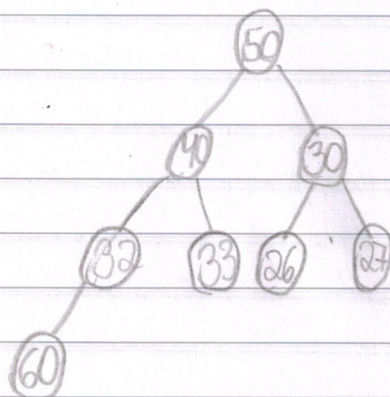
Teríamos que ampliar o vetor e refazer o processo com os números colocados no antigo vetor.

5-) É adequado manter o tamanho ocupado em no máximo 75%. 50% traz bons resultados e menos de 25% pode apresentar perda de desempenho.

6-1) Tabela de dispersão. Porque bem distribuída tendo $O(1)$, e a complexidade é inferior ao da lista encadeada.

7-1) Árvore cheia.

8-1) A complexidade é $O(\log n)$



9-1) Grafo não direcionado, as arestas não têm direção. Ex.: amizade.
 Grafo direcionado, o relacionamento é definido na direção da aresta. Ex.: Hierarquia de cargo na empresa.

10-1)

	Lista	Matriz
Esp	$O(V+E)$	$O(V^2)$
Mem	$O(V)$	$O(1)$