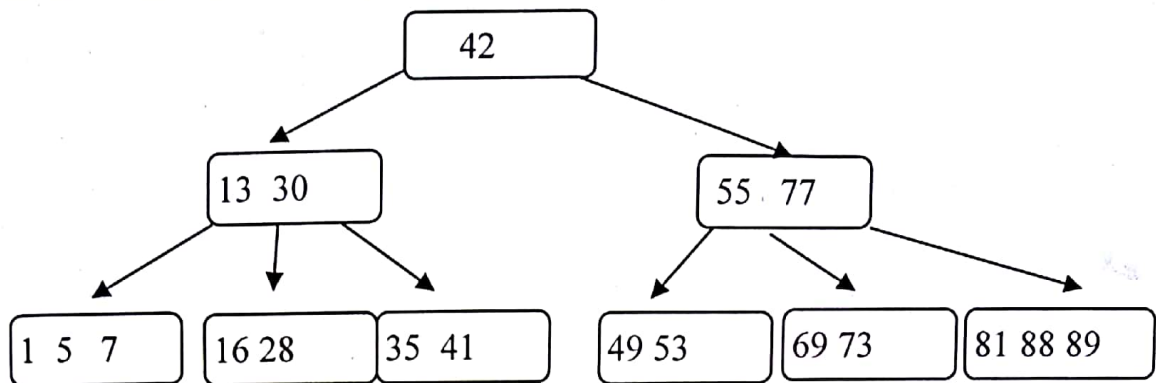


- ✓ 1.1 – Na árvore B de ordem 2 abaixo, remova a chave 42, e mostre como a árvore fica.



- ✓ 1.2– Insira, na árvore acima (que contém a chave 42), as chaves 50, 66, 70 e 75.
- ✓ 2.1- Considere uma heap binomial, sem avaliação tardia, inicialmente vazia. Insira as chaves 10, 15, 2, 8, 7, 3, 4, 11, 5, 7 e 6 nesta ordem.
- ✓ 2.2- Da heap obtida, remova o elemento de prioridade mínima.
- ✓ 3- Construa uma árvore de Huffman para a seguinte frase: “she sells sea shells”. Qual o tamanho desta expressão se codificada utilizando sua árvore? (não esqueça do espaço !)
- ✓ 4.1- Insira as seguintes chaves: 6, 41, 33, 17 e 7 em uma tabela hash com encadeamento externo. Utilize a função de hash $h(x) = x \bmod 5$.
- ✓ 4.2- Escreva o algoritmo de busca de uma chave x nesta tabela hash.
- ✓ 5- Considere 10 conjuntos inicialmente unitários, com elementos 1, 2, 3, ... 10. Realize as seguintes operações, com compressão de caminhos e união por rank.

Une(1,2), une(1,5), une(1,3), une(4,5), une(6,7), une(8,9), une(6,9), une(1,9), une(6,10). **Lembre-se que as operações de união incluem operações find (com compressão de caminhos !).**

- ✓ 6. Considere as seguintes operações:
Push(x) → insere x em uma pilha;

Pop() → retira o elemento do topo da pilha se houver;

Multi-pop(k) → retira k elementos do topo da pilha. Se a pilha tiver menos de k elementos, então retira todos os elementos da pilha.

- ✓ a) Qual a complexidade de pior caso destas operações?
b) Faça uma análise amortizada mostrando que todas estas operações têm custo amortizado constante. Explique claramente onde são colocadas moedas ou a função potencial.