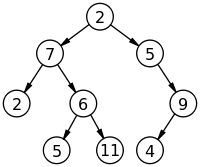
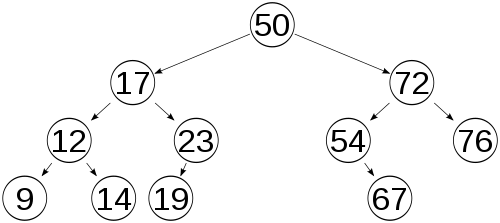
1-

**Primeiramente, destacasse os conceitos de arvore binaria e AVL.**  
  
**Árvore binária ->** É uma estrutura de dados caracterizada por:  
Ou não tem elemento algum (árvore vazia).  
Ou tem um elemento distinto, denominado raiz, com dois apontamentos para duas estruturas diferentes, denominadas sub-árvore esquerda e sub-árvore direita.  
Perceba que a definição é recursiva e, devido a isso, muitas operações sobre árvores binárias utilizam recursão. É o tipo de árvore mais utilizado na computação. A principal utilização de árvores binárias são as árvores de busca.  
Os nós de uma árvore binária possuem graus zero, um ou dois. Um nó de grau zero é denominado folha.  
Uma árvore binária é considerada estritamente binária se cada nó da árvore possui grau zero ou dois. A profundidade de um nó é a distância deste nó até a raiz. Um conjunto de nós com a mesma profundidade é denominado nível da árvore. Denominamos altura, a maior profundidade existente na árvore, ou seja, a profundidade do nó mais profundo. Uma árvore é dita completa se todas as folhas da árvore estão no mesmo nível da árvore.



**Árvore AVL ->** É uma [árvore binária de busca](https://pt.wikipedia.org/wiki/%C3%81rvore_bin%C3%A1ria_de_busca) balanceada, ou seja, uma árvore balanceada (árvore completa) são as árvores que minimizam o número de comparações efetuadas no pior caso para uma busca com chaves de probabilidades de ocorrências idênticas. Contudo, para garantir essa propriedade em aplicações dinâmicas, é preciso reconstruir a árvore para seu estado ideal a cada operação sobre seus nós (inclusão ou exclusão). Nessa estrutura de dados cada elemento é chamado de nó. Cada nó armazena uma chave e dois ponteiros, uma para a sub-árvore esquerda e outro para a sub-árvore direita.



**Segundamente, abordando as suas diferenças.**  
  
Uma árvore binária balanceada (AVL) é uma árvore binária na qual as alturas das duas sub-árvores de todo nó nunca diferem em mais de 1. Uma árvore binária de busca depende da ordem da inserção para ter um tempo assintótico de busca ótimo, visto que o primeiro valor inserido será usado como uma raiz e os demais irão para esquerda ou para direita se forem maiores ou menores. Sendo assim se você adicionar os valores em ordem crescente de s ficarão todos a direita do valor anterior, logo o tempo de busca será de O(n), sendo n o número de valores. Já em uma árvore AVL isso não ocorre, pois, cada valor na árvore possui um dado que determina seu balanceamento baseado na altura do seu nó a direita menos a altura do seu nó a esquerda, lembrando que esses valores podem ser -1=<FB<=1.  
Caso, após uma inserção qualquer valor da árvore fique com um fator de balanceamento diferente desses valores, a arvore se reestrutura mudando suas ligações para que todos os seu nós tenhamos esse fator de balanceamento. Sendo assim o tempo de busca assintótico ficará em torno de O independente da ordem de inserção dos valores.