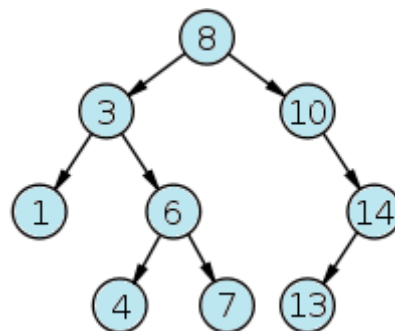


## Problema A - Árvore Binária de Busca

Em computação, a árvore binária de busca ou árvore binária de pesquisa é uma estrutura baseada em nós (nodos), onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz (e assim sucessivamente). O objetivo desta árvore é estruturar os dados de forma flexível, permitindo a busca binária de um elemento qualquer da árvore.

A grande vantagem das árvores de busca binária sobre estruturas de dados convencionais é que os algoritmos de ordenação (percurso infixo) e pesquisa que as utilizam são muito eficientes.

Para este problema, você receberá vários conjuntos de números e a partir de cada um dos conjuntos, deverá construir uma árvore binária de busca. Por exemplo, a sequência de valores: 8 3 10 14 6 4 13 7 1 resulta na seguinte árvore binária de busca:



### Entrada

A entrada contém vários casos de teste. A primeira linha da entrada contém um inteiro  $C$  ( $C \leq 1000$ ), indicando o número de casos de teste que virão a seguir. Cada caso de teste é composto por 2 linhas. A primeira linha contém um inteiro  $N$  ( $1 \leq N \leq 500$ ) que indica a quantidade de números que deve compor cada árvore e a segunda linha contém  $N$  inteiros distintos e não negativos, separados por um espaço em branco.

### Saída

Cada linha de entrada produz 3 linhas de saída. Após construir a árvore binária de busca com os elementos de entrada, você deverá imprimir a mensagem "Case n:", onde  $n$  indica o número do caso de teste e fazer os três percursos da árvore: prefixo, infixo e posfixo, apresentando cada um deles em uma linha com uma mensagem correspondente conforme o exemplo abaixo, separando cada um dos elementos por um espaço em branco.

Obs: Não deve haver espaço em branco após o último item de cada linha e há uma linha em branco após cada caso de teste, inclusive após o último.

Exemplo de entrada	Exemplo de saída
2 3 5 2 7 9 8 3 10 14 6 4 13 7 1	Case 1: Pre.: 5 2 7 In...: 2 5 7 Post: 2 7 5  Case 2: Pre.: 8 3 1 6 4 7 10 14 13 In...: 1 3 4 6 7 8 10 13 14 Post: 1 4 7 6 3 13 14 10 8