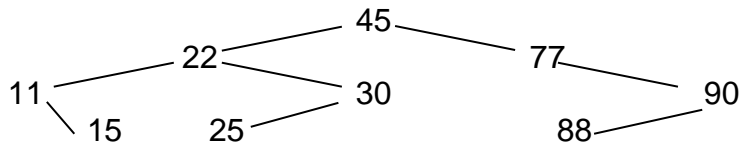


## Opgaver onsdag den 5. februar

Brug den udleverede delvise implementation af et binært søgetræ [tree.zip] i følgende delopgaver:

### Opgave 1

1. Opbyg i en App et binært søgetræ svarende til figuren herunder.



Dvs. kald metoden insert med værdierne i træet i en passende rækkefølge.

2. I hvilken rækkefølge vil data i ovenstående træ blive udskrevet ved hhv preorder, inorder og postorder gennemløb? (det skal du tænke dig til 😊)
3. Implementer de tre træ gennemløb: preorder, inorder og postorder i klassen BST, idet hvert af gennemløbene skal udskrive data fra knuderne i den angivne rækkefølge. Afprøv den på det opbyggede træ. Kommer dataene i den forventede rækkefølge?

### Opgave 2

Tilføj nedenstående metoder i det udleverede binære søgetræ. Dvs. implementer metoderne

- `inorderList()` // der returnerer en List med indholdet af alle knuderne i inorder rækkefølge.
- `height` // der returnerer højden på det binære søgetræ (skal baseres på del-løs og kombiner skabelonen)
- `findMax` – metoden skal returnere det største element i søgetræet
- `findMin` – metoden skal returnere det mindste element i søgetræet
- `leafCount()` // der returnerer antallet af blade i træet

Afprøv metoderne på træet opbygget i opgave 1.

### Opgave 3

Idet det kan antages at det binære træ indeholder Integer objekter, skal der til det binære træ tilføjes en metode, der beregner summen af indholdet i knuderne i træet. Løsningen skal basere sig på del-løs og kombiner skabelonen.

Hvad er Store O, størrelsesordenen af tidskompleksiteten for ovenstående metoder.