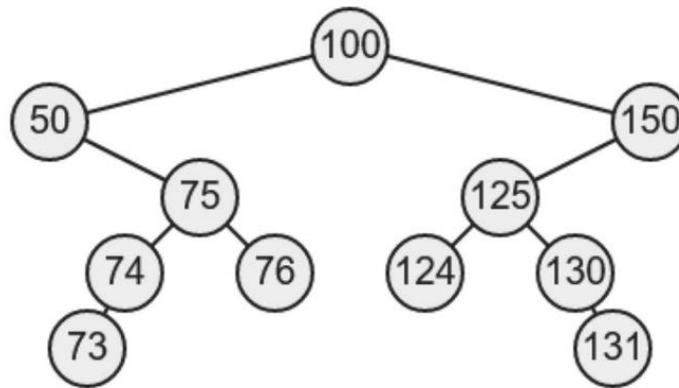


## Lista 4 zad 9

Przejście drzewa in order:



Wynik: 50,73,74,75,76,100,124,125,130,131,150

Jak widzimy takie przejście jednocześnie sortuje wartości węzłów. Aby uniknąć użycia rekurencji zastosujemy stos, w którym przechowywać będziemy już odwiedzone, ale jeszcze nie wypisane węzły

1. Tworzymy pusty stos i ustawiamy obecny na pierwszy węzeł  
X: Dopóki obecny nie jest nullem:
2. Wkładamy obecny do stosu i ustawiamy obecny na wartość obecny.lewy
3. Jeśli obecny jest nullem ale stos nie jest pusty
  - a. Wykonujemy na stosie pop i zwracamy otrzymany item.wartosc
  - b. Ustawiamy obecny na item.right
  - c. Wracamy do punktu X.
4. Jeśli obecny jest nullem i stos jest pusty skończyliśmy

### Algorytm:

```
obecny = pierwszy węzeł
stos = []
while True:
    jeżeli obecny != NULL:
        stos.dodaj(obecny)
        obecny = obecny.lewy
    w przeciwny razie jeżeli stos != NULL:
        obecny = stos.pop()
        zwróć obecny.wartosc
        obecny = obecny.prawy
    w przeciwnym razie:
        opuść pętlę
```

### Złożoność:

Nie używamy rekurencji, a przechodzimy całe drzewo jedną pętlą stąd złożoność obliczeniowa wynosi  $O(n)$  gdzie  $n$  to ilość węzłów.