

## Labbexamination - exempel

Du kan få poäng även om inte uppgifterna är helt lösta eller har den bästa lösningen. För godkänt krävs 5p. Lycka till!

### 1. Implementera från pseudokod.

Nedan finns pseudokod för partitionsdelen av quicksort. Den använder en funktion SWAP som tar en lista och två index  $i$  och  $j$ . SWAP byter plats på listans värden på index  $i$  och  $j$ . Ni får implementera SWAP, eller göra bytet direkt i PARTITION utan att anropa en funktion. Implementera partition. (5p)

**function:** PARTITION( $in\_list, lo, hi, pivot$ ) **returns**

the index of the first element of the partition containing elements higher than or equal to pivot.

**inputs:**  $in\_list$  – the list in which we partition a part of the list

$lo$  – the index in  $in\_list$  where we start partitioning. (Elements before  $lo$  in  $in\_list$  are unaffected.)

$hi$  – the index in  $in\_list$  where partitioning ends. (Elements after  $hi$  in  $in\_list$  are unaffected.)

$pivot$  – the value to compare with

**while**  $lo \leq hi$  **do**

**while**  $lo \leq hi$  **and**  $in\_list[lo] < pivot$  **then**

$lo \leftarrow lo + 1$

**while**  $lo < hi$  **and**  $pivot < in\_list[hi]$  **then**

$hi \leftarrow hi - 1$

**if**  $lo \geq hi$  **then**

**return**  $lo$

    SWAP( $in\_list, lo, hi$ )

$lo \leftarrow lo + 1$

$hi \leftarrow hi - 1$

**return**  $lo$

## 2. Gör en ändring

I klassen för det binära sökträdet, `BinarySearchTree.py` implementera en funktion `min(self)` som returnerar nyckel-värde-paret för noden som har lägst nyckelvärde. För full poäng ska funktionen ha logaritmisk tidskomplexitet om trädet är balanserat. (5p)