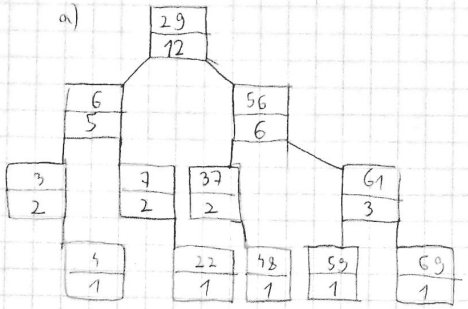
ZADATAK 1

a)




b) OS_SELECT (T.root, 8)

Funkcija vraća 8 najmanji node u stablu T, tj. vraća cvor 48. Algoritam usporeduje traženu poziciju s velicinom lijevog podstabla x i ranije pozicije korijena trenutnog podstabla(r). Ako i<r idemo lijevo, ako i>r idemo desno i oduzimamo velicinu lijevog podstabla od trazene pozicije.

c) OS_RANK (T, x) , x.key=59 Funkcija vraća rank traženog noda. Radi na principu da uzimamo rank traženog noda+1; idemo sve do korijena stabla inkrementirajući za rank lijeve grane +1 svaki put kada se nalazimo u desnoj grani.

ZADATAK 2.

```
OS_SELECT(x,i):
1  while (x != nul(T)
2      r=x.left.size+1
3      if(i==r)
4          return x
5      else if (i<r)
6          x=x.left
       else:
           x=x.right
           i=i-r
   return nul(;
```

Algoritam usporeduje traženu poziciju s velicinom lijevog podstabla x i ranije pozicije korijena trenutnog podstabla (r). Ako je i<r nda idemo lijevo, ako je i>r idemo desno i oduzimamo velicinu lijevog podstabla od tražene pozicije. Petlja se ponavlja dok ne dođemo do lista, ili dok nismo pronašli traženi element

ZADATAK 3.

```
OS_KEY_RANK (T, k)
    X = TREE_SEARCH(T.root, k)
    if(X==nul(ptr):
        return -1;
    else:
        return os_rank (T,x)

OS_RANK (T, x)
    r = x.left.size + 1
    y = x
    while y != T.root:
        if y == y.p.right
            r = r + y.p.left.size+1
        y=y.p
    return r;
```

ZADATAK 4.

a) ith Successor (Node x, int i):

```
    r = OS_KEY_RANK(x, x.key)
    return (OS_SELECT(x, i+r))
```

=>VSA: O(lgn)

b) ith Successor (Node x, int i):

```
Node y=x
for k in range (1, i+1):
    y = tree_successor (T,x)
return y
```

TEOREM IZ JEDNE PROŠLE VJ

=> O(n-1+lgn)

=> O(n)