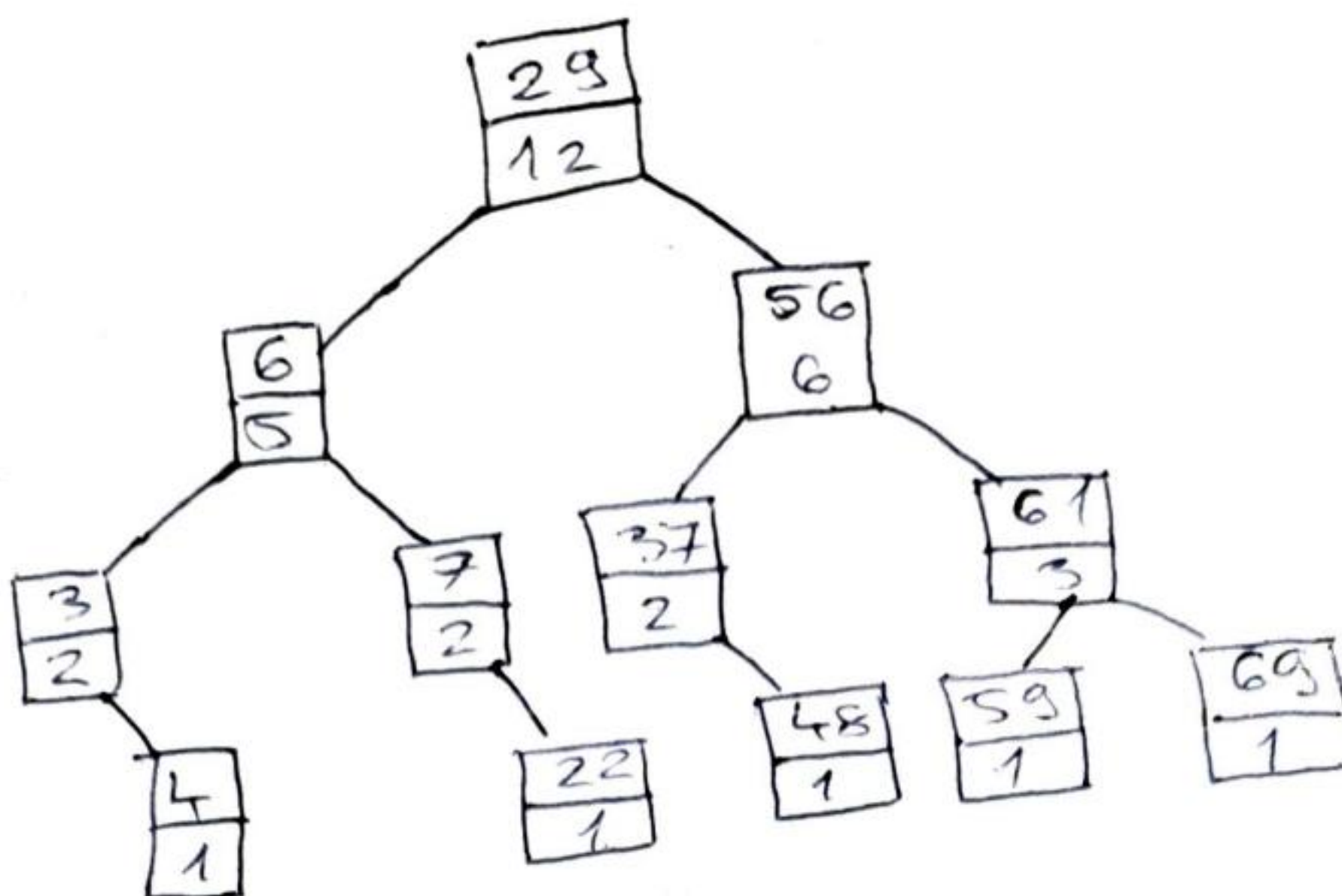


1.

1)



2) OS_SELECT(x, i)

 $r = x.\text{left.size} + 1$

if $i = r$

| return x

else if $i < r$

| return OS_SELECT(x.left, i)

else return OS_SELECT(x.right, i - r)

POSTAVLJAMO r NA VELIČINU LIJEVOG PODSTABLA + 1. GLEDAMO JE LI $i = r$, AKO JE, VRACAMO x I TO JE i -TI ČVOR. AKO NIJE $i = r$, T.J. $i < r$, ONDA TRAZIMO i -TI NAJMANJI ELEMENT U $T(x.\text{left})$ REKURZIVOM. ~~U~~ U ZADNJEJ i -TI MIN. ELEM POSTOJI U $T(x.\text{right})$ I DOBIJAMO i -TI NAJMANJI NOVE U T .

3) OS_RANK(T, x)

 $r = x.\text{left.size} + 1$
 $y = x$

while $y \neq T.\text{root}$

| if $y == y.p.\text{right}$

| | $r = r + y.p.\text{left.size} + 1$

| $y = y.p$

return r

POSTAVLJAMO r NA ISTO ŠTO I U OS_SELECT, OUDA $y = x$ I POMOĆU WHILE PETLJE IDEMO KROZ STABLO DOK NE DOĐEMO DO KORJENA. TESTIRAMO JE LI y DESNO DIJETE, AKO JE, ONDA UVEĆAMO r S VELIČINOM LIJEVOG DIJETETA RODITEJA OD $x + 1$. S TIME DOBIVAMO RANK OD x .

TAKO I ZA PRIMJER A NOVE ČJI JE KEY 59.

$r = 1 \rightarrow$ IDEMO NA RODITEJA $\rightarrow r = 4 \rightarrow$ RODITELS(56) \rightarrow

$\rightarrow r = 10 \rightarrow$ root $\rightarrow r = 10 //$


```

(2) OS_SELECT(x, u)
  while (x != NIL)
    if x.left != NIL
      | left-size = x.left.size
    else
      | left-size = 0
    if i == left-size + 1
      | return x
    else if i < left-size + 1
      | x = x.left
    else
      i = (left-size + 1)
      x = x.right
  return NIL

```

ITERIRAMO STABLOM S WHILE PETLJOM I U SVAKOM KORAKU IMA VELIČINU UJEVOG PODSTABLA OD CURRENT-NODE. USPOREĐUJEMO TU ~~VARIJABLU~~ S i - OSTALAK JE ISPITIVANJE i I $\text{left-size} + 1$. AKO JE $>$ IDEMO U UJEVO PODSTABLO, INAČE DESNO.

```

(3) OS_KEY_RANK(T, k)
  | return OS_KEY_RANK_HELP(T, T.root, k)
OS_KEY_RANK_HELP(T, x, k):
  if x == NIL
    | return 0
  if x.key == k
    | return (1 + x.left.size)
  if x.key > k
    | return OS_KEY_RANK_HELP(T, x.left, k)
  return 1 + x.left.size + OS_KEY_RANK_HELP(T, x.right, k)

```