

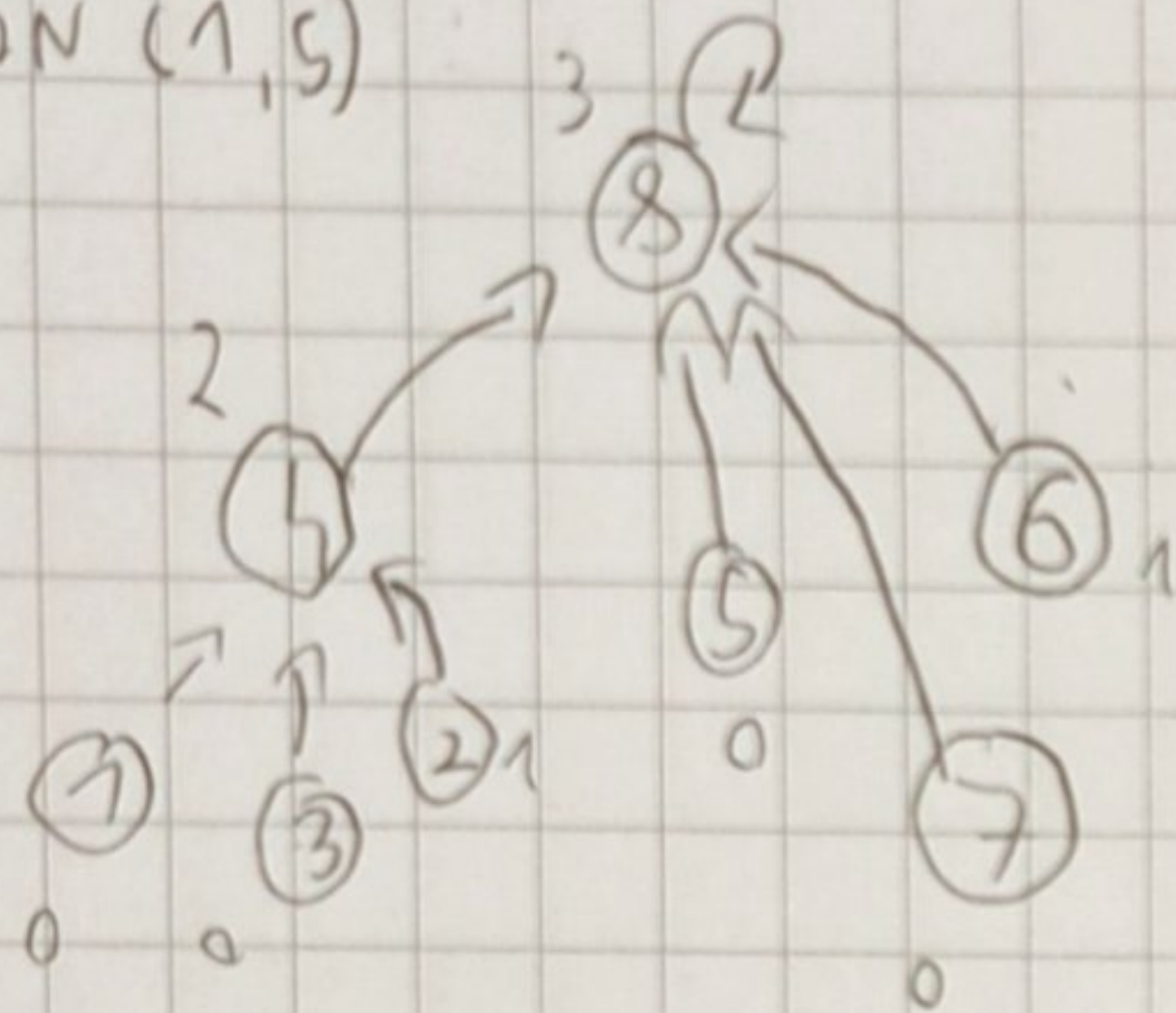
ZADACI 7

1. ZADATAK.

```

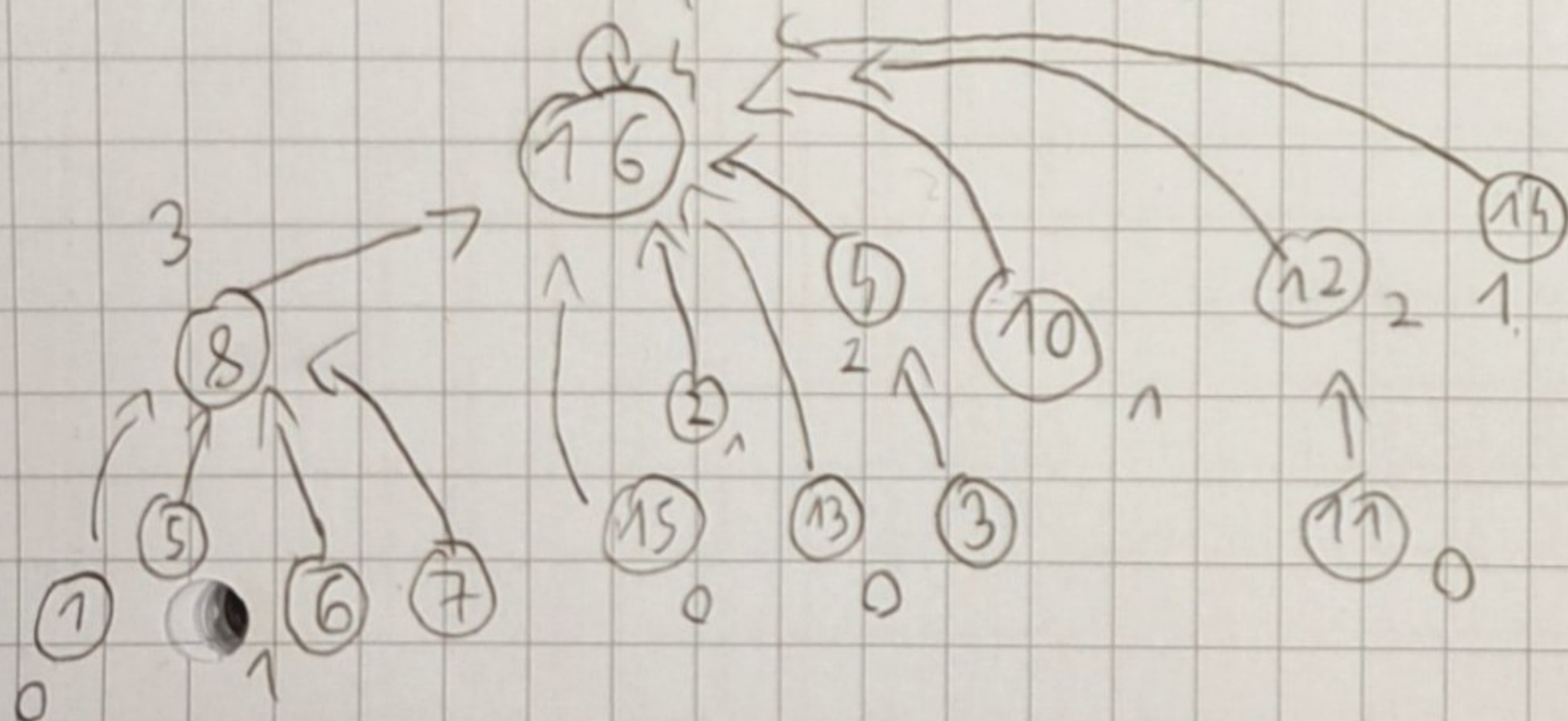
FIND_SET(x, s):
    vector<node> A
    A.push_back(x)
    y = x
    while(y != s.root)
        y = y.p
    A.push_back(y)
    for node in A:
        node.p = y
    
```

UNION(1, 5)



FIND_SET(2)

FIND_SET(9)

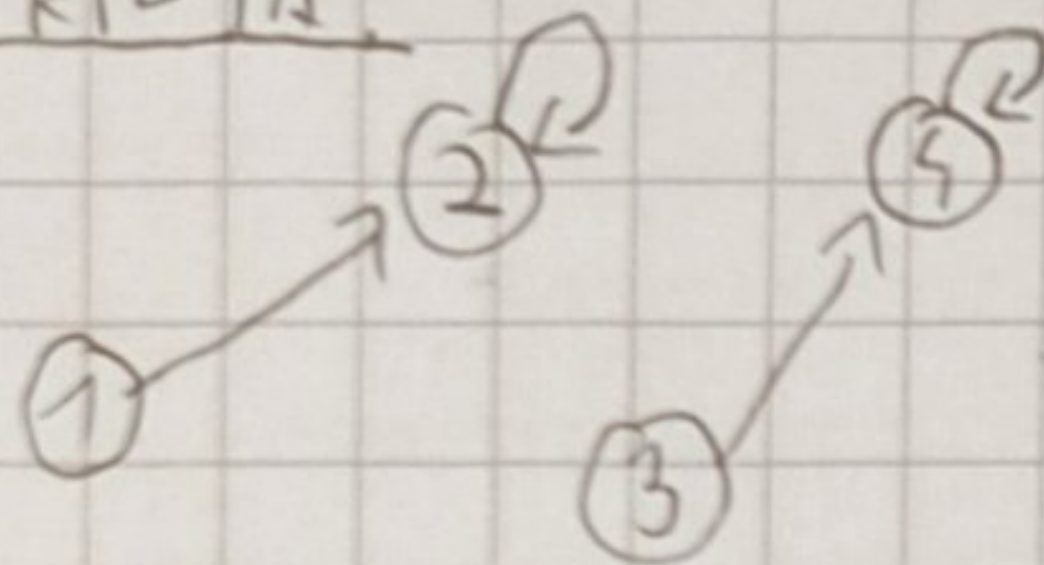


2. ZADATAK. MAKE-SET: stvar. jedinične skupove

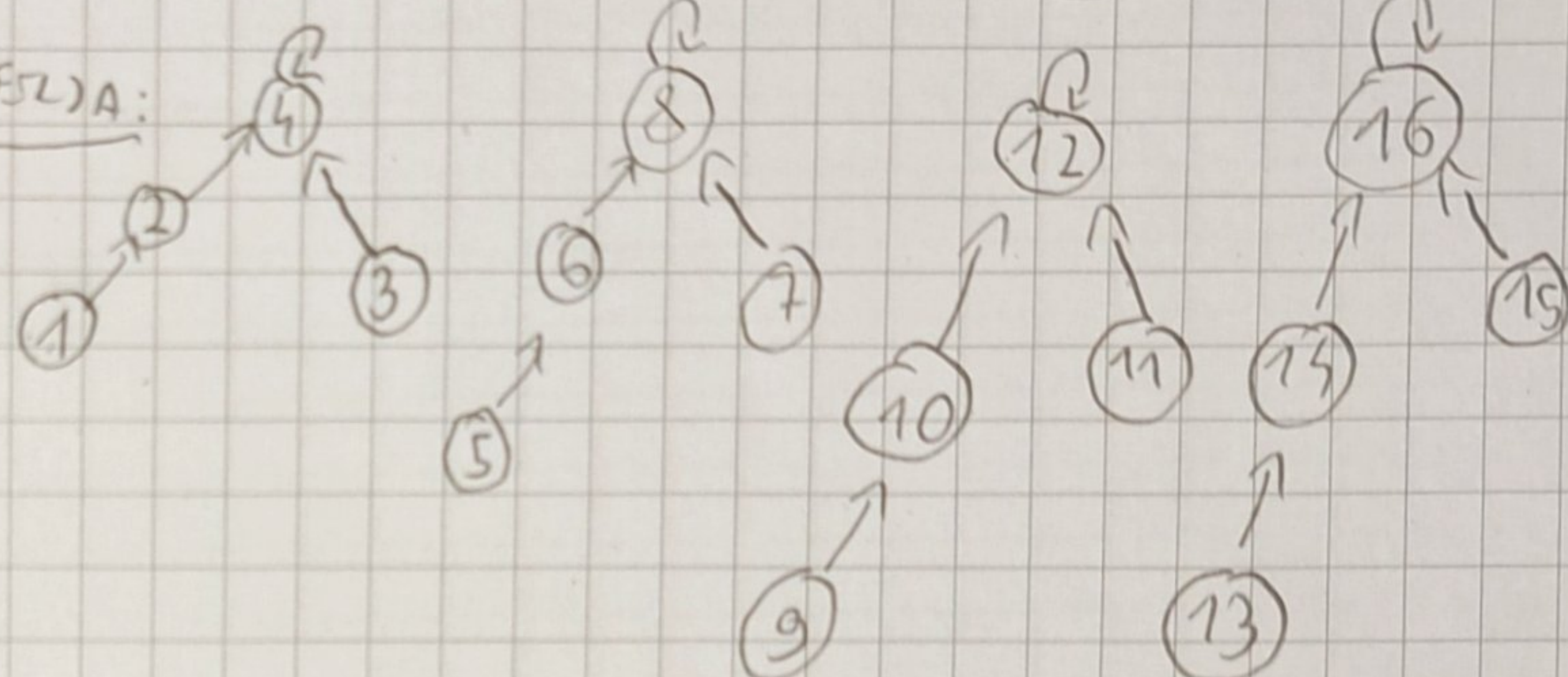
1

16

1. PETA: A:



2. PETA: A:



3. ZADATAK.

MAKE-SET kreira n skupova, $O(n)$

1) BEST-CASE UNIONA POVEĆAVA VISINU ZA 1
SVO ZNAČI DA KALON M POZIVA VISINA JE $\lg m$

2) WORST-CASE FIND-SET JE POGLAZAT NA 2
SVE KODOVE U STABLU. Ako JE STABLO
LINEARNO $\Rightarrow O(mn)$

$\Rightarrow 1) + 2) \Rightarrow O(m \lg n)$

5. ZADATAK.

Za svakog člana skupa, njegova prva polje koje je prije pohrizalo na objekt skupa sada pohrizuje na zadnji element povezane liste. Tada za bilo koji skup, njegov se zadnji element može pronaći odlaškom na početak liste; sljedeći pohrizivač kojeg taj objekt održava na zadnjeg elementa povezane liste. To zahtijeva samo praćenje tačno dva pohrizivača pa to košta konstantu. Budući da je predstavnih skupa zadnji element u skupu, kada spojimo dvije povezane liste, manji od dva skupa stavljamo ispred većeg, budući da moramo ažurirati njihove pohrizivače predstavnih skupa, za razliku od izravne situacije, gdje smo ažurirali predstavnih objekta koji su stavljani na kraj povezane liste.