

zad.1.

FIND-SET(x)

root = x

while root.p ≠ root
root = root.p

while x ≠ root

temp = x.p

x.p = root

x = temp

return root

zad.2.

Na početku imamo 16 skupova od kojih svaki sadrži i

- nakon druge for petlje imamo:

$\{1,2\}, \{3,4\}, \{5,6\}, \{7,8\}, \{9,10\}, \{11,12\}, \{13,14\}, \{15,16\}$

- nakon 3. for petlje imamo:

$\{1,2,3,4\}, \{5,6,7,8\}, \{9,10,11,12\}, \{13,14,15,16\}$

- UNION(1,5) vraća

$\{1,2,3,4,5,6,7,8\}, \{9,10,11,12\}, \{13,14,15,16\}$

- UNION(11,13) vraća: $\{1, \dots, 8\}, \{5, \dots, 16\}$

- UNION(1,10)
vraća:

$\{1, \dots, 16\}$

FIND-SET(2) i

FIND-SET(9) vraćaju
odgovarajuće naci

zad.3.

Pretp. da je $n' = 2^k$ najmanja potencija od 2 manja od n

Da bi to vidjeli taj niz operacija uzima potrebno vrijeme

Prvo trebamo znati da nakon svake iteracije for petlje s indeksom j, imamo elemente x_1, \dots, x_n ,

koji su stablima dubine i. Kada završimo vanjsku petlju, imamo da x_1, \dots, x_n

leže u istom skupu, ali su predstavljeni stablom dubine $k \in \Omega(\lg n)$

Budući da uzastopno pozivamo FIND-SET na stavku koja je lgn udaljena od svog predstavnika skupa,

imamo da je svakom potrebno vrijeme lgn. Dakle, za posljednu for petlju ukupno je potrebno $\Omega(n \lg n)$

zad.4.

MAKE-SET(x)

o → objekt s tri polja, next, value, set

L → objekt povezane liste s head = tail = o

o.next = NIL

o.set = L

o.value = x

return L

FIND-SET(x)

return o.set.head.value

UNION(x, y)

L1 = x.set

L2 = y.set

L1.tail.next = L2.head

z = L2.head

while z.next ≠ NIL do

z.set = L1

end while

L1.tail = L2.tail

return L1

zad.5.

Za svaki član skupa napraviti ćemo njegovo prvo polje koje je prije pokazivalo nazad na točku skupa objekta umjesto na zadnji element povezane liste. Zatim, s obzirom na bilo koji skup, možemo pronaći njegov zadnji element tako da odemo na head i pratimo pointer koji taj objekt održava na zadnji element povezane liste. To samo zahtijeva praćenje točno dva pointera tako da je potrebno konstantno vrijeme. Bitno je pripaziti prilikom spajanja tih modificiranih skupova. Budući da je predstavnik zadnji element skupa, kada spojimo 2 povezane liste, spremimo manji od 2 skupa prije većeg, jer trebamo ažurirati njihove predstavljajuće pointere, za razliku od originalne situacije, gdje moramo ažurirati predstavnike objekata koji su postavljeni na kraj povezane liste.