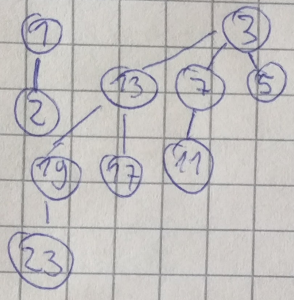
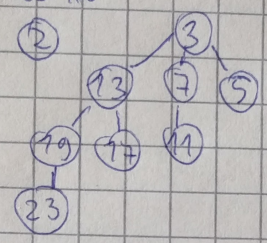
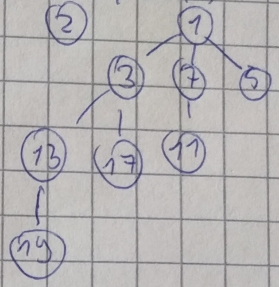
1) a) INSERT 1:



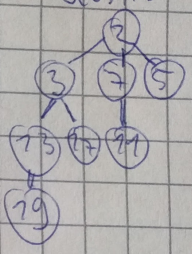
DEL MIN



DECR. KEY (23, 1)

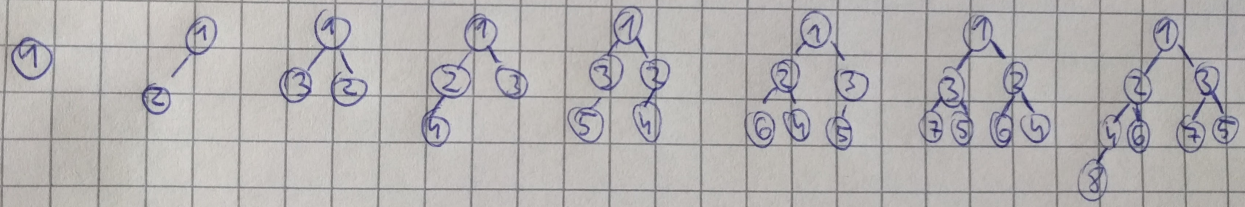


DEL MIN



b) OPIS ZLIVANJA V PRILOGI

- VSTAVLJANJE:



c) RAZLIKA IN FUNKCIJE V PRILOGI

- BOOLEAN GETLAYER (INT A) – V PRILOGI

```
1. b)
    Opis operacije zlivanja skrivljenih kopic:
        opisal bom nerekurzivno zlivanje saj je lažje za pojasniti:
        1. razrežemo drevo od korena, tako da je vsak element od korena desno sam
        (tako da ima root samo leve otroke ali pa je brez otrok)
        2. uredimo drevesa v naraščajočem vrstnem redu glede na njihov root ključ
        3. združujemo drevesa od največjega (od desne) proti najmanjšim (proti levi)


    personal zapiski za skew insert (da jih imam na kupu)
        pogleda v korenu v katero podvejo spada
        ko prideš v list:
             -switcha lev in desn poddrevo
             - to dela rekurzivno do root (v rootu tud switcha lev in desn poddrevo)

1. c)
    MIN MAX heap -> razlika od običajne dvojiške kopice
    razlika:
        -je implicitna podatkovna struktura
        -find je v konstantnem času namesto O(n)
        -vsak node na lihem nivoju je manjši kot otroci, vsak na sodem pa večji
    funkcije:
                 worst:
        insert   O(lgn)
        delete   O(lgn)
        build    O(n)
        find     O(1)
        delMin   O(lgn)


    Koda je v prilogah
    njena časovna zahtevnost za poljuben i je pa O(1) saj vse kar naredi je
        da izračuna logaritem števila ter ga z modulom zdeli.
```

```python
import math
def get_layer(i):
    if i == 0:
        nivo = 1
    elif i == 1:
        nivo = 2
    else:
        nivo = math.ceil(math.log(i+2, 2))
    return str(nivo)+" max" if nivo %2 == 0 else str(nivo)+" min"

def main():
    for i in range(10):
        nivo = get_layer(i)
        print(nivo)
main()
```

rezultati

```
1 min
2 max
2 max
3 min
3 min
3 min
3 min
4 max
4 max
4 max
```