

Naloga 3.2

Skupina n otrok se postavi v vrsto in se igra izštevanko. Če ima izštevanka m zlogov, otrok na m -tem mestu izpade in izštevanka se ponovno začne pri otroku $m + 1$. Če pridemo do konca vrste, začnemo ponovno od začetka. Izštevanko ponavljamo toliko časa, dokler ne izpade zadnji otrok. Zanima nas vrstni red izpadanja otrok glede na njihovo zaporedno številko v začetni vrsti. Na primer, če ima izštevanka $m=3$ zloge in imamo $n=10$ otrok, potem izpade najprej 3. otrok, potem 6., 9., 2., 7. itd.

A) Zapišite vrstni red izpadanja za $m=6$, $n=20$

Vrstni red je: 6 12 18 4 11 19 7 15 3 14 5 17 10 8 2 9 16 13 1 20.

B) Napišite psevdokodo čimbolj učinkovitega algoritma izštevanja, ki vrne zaporedje izpadlih za poljubni m in n . Ocenite časovno in prostorsko zahtevnost algoritma glede na n in m .

```
Function izloči(int m,int n){
    Int[] Otroci = new int[n];
    Int[] izloceni = new int[n];
    m-=1;
    int j = 0;
    i = m;
    while(otroci.length() > 1){
        izloceni[j] = otroci[i];
        i = (m+i)%otroci.length();
        j++;
    }
    Izloceni[j]=otroci[0];
    Return izloceni;
}
```

Časovna Zahtevnost je $O(n)$ ker moramo čez celo tabelo otrok z 1 zanko
prostorska pa tudi $O(2n)$ ker na koncu imamo 2 tabeli po $2n$ elementov

C) Sedaj spremenimo pravila igre: izpadli otrok izbere izštevanko za naslednje izštevanje (torej izbere m). Napišite psevdokodo čimbolj učinkovitega algoritma izštevanja za dani n in spremenljivi m . Ocenite časovno in prostorsko zahtevnost algoritma v odvisnosti od n in m .

```
Function izloči(int m,int n){
    Int[] Otroci = new int[n];
    Int[] izloceni = new int[n];
    m-=1;
    int j = 0;
    i = m;
    while(otroci.length() > 1){
        izloceni[j] = otroci[i];
        m = random.nextInt(0,10);
        i = (m+i)%otroci.length();
        j++;
    }
    Izloceni[j]=otroci[0];
    Return izloceni;
}
```

Vse je isto samo da vsakič posebj m nastavimo na novo. To sem kar impementiral s fucnkcijo random ki ponazarja random odločitev otroka. Časovna zahtevnost je $O(n)$ ker moramo iti čez vse otroke v vsakem primeru. Prostorka je pa $O(2n)$ ker imamo nakoncu 2 tabeli po n elementov