



## Rešenja zadataka

**Zadatak 1.** Rešenje je pravolinijsko: potrebno je sortirati niz prema zadatim kriterijumima, pri čemu za svakog učenika pamtimo  $ind[i]$  - njegov redni broj u početnom (nesortiranom) nizu. Sada je traženi učenik onaj čija je vrednost  $i - ind[i]$  najveća, a ukoliko ima više takvih, rešenje je učenik sa najmanjim indeksom.

Ukoliko se koristi QuickSort, složenost je  $O(n \cdot \log n)$ .

**Zadatak 2.** Za svakog igrača  $i$  je potrebno pronaći najveći indeks  $j$  takav da je suma niza  $c$  od  $t[i]$ -tog do  $j$ -tog mesta manja ili jednaka od  $p[i]$ , i tada je za njega rešenje  $j - t[i] + 1$ . Ukoliko u pomoćnom nizu  $a[k]$  pamtimo sume od prvog do  $k$ -tog elementa niza  $c$ , problem se svodi na nalaženje najvećeg indeksa  $j$  niza  $a$  na segmentu  $[t[i], n]$  tako da je  $(a[j] - a[t[i] - 1]) \leq p[i]$ . Ovo radimo binarnom pretragom što nas dovodi do složenosti  $O(m \cdot \log(n))$ .

**Zadatak 3.** Potrebno je primetiti (što sugeriše i uslov broj 3.) da se Boži ne isplati da ima previše šifara. Neka je  $d$  - dužina najkraćeg puta od  $u$  do  $v$  (po broju grana) i  $p$  jedan takav put. Ako bi bilo  $k > d$  onda mora postojati šifra koja se ne nalazi ni na jednoj grani puta  $p$  (prema uslovu 1.). Ako Žika provali sve šifre osim nje, on može putem  $p$  da dođe do Božinog kompijutera, što je u suprotnosti sa uslovom 2.

Prema tome  $k \leq d$ . Lako se pokazuje da  $k = d$  zadovoljava sve uslove (obilazimo graf *BFS*-om i granama do najbližih čvorova dodelimo šifru 1, sledećim šifru 2 itd.). Ukoliko je  $d = \infty$ , rešenje je očigledno  $m$ . Ukupna složenost algoritma je složenost *BFS*-a što je  $O(n + m)$ .

**Zadatak 4.** Zadatak radimo dinamičkim programiranjem - neka je  $d[i]$  broj načina na koji možemo u niz poređati  $i$  povrća. Za  $i \leq k + 1$ , rešenje je  $i + 1$ . Neka je  $i > k + 1$ . Ako je poslednji u nizu paradajz, preostali deo niza možemo izabrati na  $d[i - 1]$  načina. Ukoliko je poslednji u nizu kupus, na preposlednjih  $k$  mesta se moraju nalaziti paradajzi, dok preostali deo niza biramo na  $d[i - k - 1]$  načina. Dakle  $d[i] = d[i - 1] + d[i - k - 1]$ . Iz rekurentnog uslova vidimo da je složenost algoritma linearna.

**Zadatak 5.** Ključna stvar je primetiti da dužina šifre ne može biti duža od  $\log n$ . Zaista, broj štringova (koji su kandidati za šifru) dužine  $k$  jednak je  $2^k$ . U datom nizu se pojavljuju najviše  $n - k + 1$  štringova dužine  $k$ . Ukoliko bi dužina šifre bila veća od  $k$  (tj. svi štringovi dužine  $k$  se pojavljuju u nizu) onda bi važio da je  $2^k < n$ , tj.  $k < \log(n)$ . Prema tome, treba ispitati svega  $\log n$  različitih dužina (počevši od najmanje).

Kako je  $\log n < 20$ , za fiksirano  $k$  prolazimo kroz generisani string i markiramo svaki podstring dužine  $k$  koji se u njemu pojavljuje (posmatramo znak 'a' kao nulu a 'b' kao jedinicu, pa konvertujemo podstring dužine  $k$  u broj) u složenosti  $O(n)$ . Zatim za svaki od  $2^k = O(n)$  stringova ispitujemo da li je markiran. Na ovaj način do šifre dolazimo u složenosti  $O(n \cdot \log(n))$ .