# OBLIKOVANJE I ANALIZA ALGORITAMA — 2. kolokvij

## 9. 1. 2013.

1. (20) Mala kopiraonica ima samo jedan kopirni aparat i želi optimizirati zadovoljstvo svojih dnevnih korisnika. Svako jutro, pri otvaranju, poznat je skup od $n$ poslova koje treba obaviti taj dan. Posao za $i$-tog korisnika troši $t_i$ vremena kopirnog aparata. Osim toga, svaki korisnik ima poznatu težinu $w_i$, koja predstavlja važnost tog korisnika za cjelokupni posao.

Zadovoljstvo svakog korisnika ovisi o trenutku **završetka** njegovog posla — "što prije, to bolje". Za bilo koji poredak poslova, neka $C_i$ označava trenutak završetka posla za $i$-tog korisnika. Na primjer, ako je posao za $i$-tog korisnika obavljen kao **prvi** po redu, onda je $C_i = t_i$, a ako je posao za $i$-tog korisnika obavljen **odmah** iza posla za $j$-tog korisnika, onda je $C_i = C_j + t_i$, tj. aparat radi bez prestanka i vremena se zbrajaju.

Kopiraonica želi **poredati** poslove tako da **minimizira** težinsku sumu vremena završetka poslova

$$S = \sum_{i=1}^{n} w_i C_i.$$

Na primjer, neka je $n = 2$. Posao za prvog korisnika traje $t_1 = 1$, s težinom $w_1 = 10$, a posao za drugog korisnika traje $t_2 = 3$, s težinom $w_2 = 2$. Ako posao za prvog korisnika obavimo ranije, onda je pripadna težinska suma vremena završetka jednaka $10 \cdot 1 + 2 \cdot 4 = 18$, dok u obratnom poretku dobivamo veću težinsku sumu vremena završetka $10 \cdot 4 + 2 \cdot 3 = 46$.

(a) Analizirajte slučaj $n = 2$ i nađite **pohlepni** kriterij koji daje najmanju težinsku sumu $S$. Dokažite **optimalnost** tog kriterija za bilo koji $n$.

(b) Sastavite algoritam koji nalazi optimalni poredak poslova i nađite njegovu složenost. Ulazni argumenti su broj $n$ i polja $t$, $w$. Izlaz algoritma su permutacija $p$ brojeva od 1 do $n$, koja sadrži optimalni redosljed poslova ($p[j] = i$ znači da posao za $i$-tog korisnika treba obaviti kao $j$-ti po redu), i pripadna najmanja težinska suma $S$. Složenost algoritma mora biti $O(n^2)$.

**OKRENITE!**

2. (30) Svaka od $n$ osoba zna točno **jednu** glasinu, različitu od svih ostalih glasina. Cijela skupina želi što brže podijeliti "novosti" sa svim ostalim osobama, međusobnim slanjem poruka. Svaka pojedina poruka sadrži **sve** glasine koje pošiljatelj zna u trenutku slanja poruke i ima samo **jednog** primatelja. Slanje jedne poruke osobe $i$ osobi $j$ zapisujemo kao poziv funkcije *poruka*($i$, $j$). Zbog proizvoljne numeracije osoba, uzmite da prvu poruku šalje osoba 1, a zadnju šalje osoba $n$.

(a) Nađite **najmanji** ukupni broj poruka koje ove osobe trebaju poslati, tako da bude sigurno da je **svaka** osoba saznala **sve** glasine. Dokažite da je to najmanji mogući broj poruka koji garantira "potpuno širenje" glasina. Napišite neki algoritam — redosljed slanja poruka, koji to realizira u najmanjem broju poruka.

U nastavku zadatka tražimo posebne algoritme za "potpuno širenje" glasina, koji imaju **najmanji** ukupni broj poruka i **još** zadovoljavaju dodatna ograničenja.

(b) Pretpostavimo da svaka glasina ima istu "duljinu" u poruci — na primjer, jednaku 1. Napišite algoritam koji ima **najmanji** ukupni **zbroj** duljina svih poruka i dokažite njegovu optimalnost.

(c) Svakom poslanom porukom želimo **maksimalno povećati** ukupan **zbroj** poznatih glasina po svim osobama. Na početku, ovaj zbroj je $n$ — svaka osoba zna jednu glasinu, a na kraju je $n^2$ — svaka osoba zna svih $n$ glasina. Napišite "pohlepni" algoritam koji to realizira i dokažite njegovu optimalnost.

Napomena: Ako ste pod (b) ili (c) napisali odgovarajući algoritam, nije potrebno posebno pisati neki algoritam pod (a).

3. (20) Zadano je polje od $n$ objekata. Te objekte možemo uspređivati **binarnom** uređajnom relacijom $\leq$. Definirajte što je problem sortiranja takvog polja i kako mjerimo složenost.

(a) Opišite ukratko Quicksort algoritam za sortiranje polja.

(b) Napišite **rekurziju** za složenost (ili neku mjeru složenosti) Quicksort algoritma. Što je rješenje te rekurzije u **najgorem** slučaju i kad se to događa, tj. kako tad izgleda polazno polje?

(c) Ukratko opišite uz koje pretpostavke se izvodi **prosječna** složenost i koji rezultat se dobiva tom analizom.

(d) Kolika je **donja** ograda složenosti za **bilo koji** algoritam sortiranja polja koji koristi samo binarnu operaciju uspoređivanja elemenata u polju? Ukratko argumentirajte kako dolazimo do te ograde.

**REZULTATI:** petak, 11. 1. 2013. u 12 sati. Saša Singer