Kopiranje – Rješenje

Algoritam je sljedeći:

1) Uzmemo neki broj S koji nam predstavlja najveću sumu elemenata neke grupe
2) Provjerimo da li je moguće napraviti K grupa tako da im suma ne bude veća od S
3) Ako jeste, pokušamo sa manjim brojem S. A ako nije, pokušamo sa većim.
4) Nakon što smo našli najmanji broj S za koji je moguće naći K grupa tako da im suma elemenata ne bude veća od S, konstruišemo te grupe.

Korak 2) je tipični greedy algoritam. Realizira se tako što prolazimo petljom kroz elemente niza i sabiramo sve elemente dok ne dođemo do kraja ili do nekog elementa čijim bi sabiranjem suma postala veća od S. Time smo našli jednu moguću grupu. Ukoliko je trenutni broj nađenih grupa jednak K a nismo došli do kraja, onda nije moguće napraviti K grupa tako da suma njihovih elemenata nije veća od S i javimo da to nije moguće. U suprotnom, krećemo iznova sabiranje od sljedećeg elementa. Ukoliko se petlja završila bez poruka o grešci, javimo da je trenutno S moguće. Vremenska složenost je $O(N)$.

Korak 4) je skoro identičan koraku 2). Sada trebamo konstruisati grupe na osnovu datog rješenja S. Sa obzirom da trebamo minimizirati broj elemenata u prvoj grupi, pa u drugoj itd. ovdje ćemo sabiranje vršiti od kraja tako da zadnja grupa bude što je moguće veća itd. Nakon što saberemo najviše elemenata što možemo, označimo da na tom mjestu treba biti znak '/' u izlazu i nastavimo sa novim sabiranjem. Nakon što se ta petlja završi, ispišemo niz sa pregradama. Vremenska složenost je $O(N)$.

Rješenje S koje dobijemo je minimum iz podskupa prirodnih brojeva čiji elementi zadovoljavaju svojstvo da je moguće napraviti K grupa u početnom nizu tako da suma elemenata nijedne grupe ne bude veća od S. Taj bi broj mogli tražiti i petljom – krenemo od vrijednosti najvećeg elementa u nizu (jer suma mora biti najmanje tolika) i povećavamo za jedan sve dok korak 2) vraća negativan odgovor. Prvi broj za koji je odgovor pozitivan je rješenje. Sa obzirom da je niz odgovora koraka 2) za prirodne brojeve neopadajući niz ukoliko negativan odgovor posmatramo kao „0“ a pozitivan kao „1“ (taj niz bi izgledao 0, 0, 0, ..., 0, 1, 1, 1, 1, ..., gdje se prvi broj „1“ javlja na mjestu S), tj. elementi niza su sortirani, možemo koristiti i binarnu pretragu da lociramo najmanju poziciju čiji je element jednak 1. Granice pretrage bi bile od vrijednosti najvećeg elementa niza do sume svih elemenata, što daje $O(logSUMA)$ vremensku složenost za traženje broja S, gdje je SUMA suma svih elemenata niza (ukoliko je $K = 1$ tada imamo samo jednu grupu sa svim elementima).

Sve ukupno, imamo algoritam složenosti $O(N \cdot logSUMA)$.