

Uvod u algoritme i strukture podataka

11. čas

1. (LCS) Neka su date dve niske s i t . Potrebno je pronaći najdužu sekvencu karaktera (ne nužno uzastopnih) koja se javlja u obe niske. Sa standardnog ulaza se unose dve niske s i t redom. Na standardni izlaz ispisati jednu vrednost koja predstavlja dužinu najduže zajedničke sekvence za ove dve niske.
2. (podniska) Neka su date dve niske S i T . Potrebno je odrediti na koliko načina je moguće podnisku T pronaći u niski S pri čemu karakteri ne moraju biti uzastopni. Na primer podniska *bag* se u niski *bagbag* pojavljuje 5 puta.
3. (edit) Dve su niske s i t . Potrebno je da pronaći minimalan broj operacija (umetsanje, brisanje ili zamena jednog karaktera) da bi se niska s transformisala u nisku t . Svaka operacija ima cenu 1. Sa standardnog ulaza se unose 2 niske s i t . Na standardni izlaz je potrebno ispisati jednu vrednost koja predstavlja minimalan broj operacije koji je potreban kako bi se niska s transformisala u nisku t .
4. (LIS) Neka je niz A koji se sastoji od n celih brojeva. Potrebno je odrediti dužinu najdužeg strogo rastućeg podniza (Longest Increasing Subsequence) niza A . **Elementi ne moraju biti uzastopni.** Sa standardno ulaza se unosi vrednost n , a zatim se unosi n elemenata niza. Na standardni izlaz ispisati jednu vrednost koja predstavlja dužinu najdužeg strogo rastućeg podniza niza a .
5. (vraćanje kusura) Prodavačica u prodavnici treba vrati kusur mušteriji. Na raspolaganju ima samo kovanice različitih vrednosti. Pomozimo prodavačici da vrati kusur uz korišćenje što manjeg broja kovanica. Sa standardnog ulaza se unosi broj različitih dostupnih kovanica i vrednost kusura. Nakon toga se unose vrednosti dostupnih kovanica u rastućem poretku. Na standardni izlaz ispisati koliko je minimalno kovanica potrebno za vraćanje kusura i dodatno koliko koje kovanice je upotrebljeno u ovu svrhu.
Primer:
Ulaz: 6 586
1 2 5 50 100 500
Izlaz: 10
1 0 7 1 0 1
Objašnjenje: Optimalno je iskoristiti jednu kovanicu od 500, jednu od 50, 7 od 5 i jednu od 1 kako bi se vratio kusur u vrednosti od 586.
6. (kupovina) Neka je dato n artikala, svaki sa cenom $p[i]$. Neka je dat i budžet B . Cilj je kupiti što je moguće veći broj artikala, tako da njihova ukupna cena ne prelazi B . Sa standardnog ulaza se unose vrednosti n i B . Nakon toga se unose cene za svaki od n artikala. Na standardni izlaz ispisati maksimalan broj artikala koje je moguće kupiti tako da njihova ukupna cena ne prevazilazi budžet B .