

Spisak zadataka

Čas 1.2, 1.3

Problem (minimum niza): Definirati funkciju koja određuje minimum nepraznog niza brojeva i dokaži njenu korektnost. Razmotriti rekursivnu i iterativnu implementaciju.

Problem (Euklidov algoritam): Definirati funkciju koja izračunava najveći zajednički delilac dva neoznačena broja i dokaži njenu korektnost. Razmotriti rekursivnu i iterativnu implementaciju.

Problem (određivanje binarnih cifara u zapisu broja): Definirati algoritam koji određuje binarne cifre u zapisu datog neoznačenog broja n i dokaži njegovu korektnost. Razmotriti rekursivnu i iterativnu implementaciju.

Problem (obrtnanje redosleda cifara u zapisu prirodnog broja): Neka je rev funkcija koja prirodni broj n preslikava u broj koji se zapisuje istim ciframa kao broj n , ali u obratnom poretku. Npr. $rev(1234) = 4321$, $rev(100) = 1$. Podrazumevamo da je $rev(0) = 0$. Definirati iterativni algoritam koji određuje vrednost $rev(n)$ i dokaži njenu korektnost.

Problem (holandska trobojka): Definirati algoritam koji efikasno reorganizuje elemente niza brojeva tako da prvo idu svi njegovi negativni članovi, zatim sve nule i na kraju svi pozitivni članovi. Redosled unutar grupe negativnih i unutar grupe pozitivnih članova nije bitan. Dokaži njegovu korektnost.

Problem (binarna pretraga prelomne tačke): U nizu se nalaze prvo neparni, pa zatim parni celi brojevi. Definirati funkciju koja pronalazi poziciju prvog parnog broja (ako takvog elementa nema, funkcija vraća dužinu niza). Razmotriti rekursivnu i iterativnu implementaciju.

Problem (quick-sort particionisanje): Zadatak paricionisanja u algoritmu *QuickSort* je da rasporedi elemente tako da prvo idu oni koji su manji ili jednaki pivotirajućem elementu, zatim pivot, i na kraju oni koji su strogo veći od pivota. Definiši algoritam kojim se može vršiti particionisanje dela niza određenog intervalom pozicija $[l, d]$ za $0 \leq l \leq d < n$, dokaži njegove invarijante i korektnost.

Problem (najmanji ne-zbir podskupa): Dat je skup prirodnih brojeva (zadat u obliku sortiranog niza). Definirati funkciju koja određuje najmanji prirodan broj koji se ne može dobiti kao zbir nekog podskupa elemenata tog skupa i dokazati njenu korektnost.

Problem (sortiranje umetanjem): Definirati funkciju koja sortira niz korišćenjem algoritma sortiranja umetanjem i dokaži njenu korektnost. Razmotriti iterativnu i rekursivnu varijantu.

Problem (sortiranje selekcijom): Definirati funkciju koja sortira niz korišćenjem algoritma sortiranja selekcijom i dokaži njenu korektnost. Razmotriti

iterativnu i rekurzivnu varijantu.

Čas 2.1, 2.2, 2.3

Problem (zbir prvih n prirodnih brojeva): Definirati efikasan algoritam koji izračunava zbir $1 + 2 + \dots + n$. Proceni mu složenost.

Problem (nedostajući broj): Dat je niz od n elemenata koji sadrži različite brojeve iz skupa $0, 1, 2, \dots, n$ (tačno jedan broj nedostaje). Odredi koji broj nedostaje.

Problem (broj deljivih u intervalu): Definirati efikasan algoritam koji određuje koliko je brojeva u intervalu $[a, b]$, $0 \leq a \leq b$ deljivo datim brojem k . Proceni mu složenost.

Problem (ispitivanje da li je broj prost): Definirati efikasnu funkciju koja proverava da li je broj prost. Proceni joj složenost.

Problem (Eratostenovo sito): Definirati efikasnu funkciju koja za sve brojeve manje od datog određuje da li su prosti i njenim korišćenjem izračunava koliko ima prostih brojeva manjih od datog. Proceni joj složenost.

Problem (Euklidov algoritam): Proceni složenost Euklidovog algoritma za određivanje NZD.

Problem (svi faktoriјeli): Napisati program koji ispisuje vrednosti svih faktoriјela od 1 do n .

Problem (maksimalni zbir segmenta): Definirati efikasnu funkciju koja pronalazi najveći mogući zbir segmenta (podniza uzastopnih elemenata) datog niza brojeva. Proceni joj složenost.

Problem (broj rastućih segmenata): Dat je niz a celih brojeva. Definirati efikasan algoritam kojim se određuje koliko u tom nizu postoji rastućih segmenata (rastući segment čine uzastopni elementi niza $a_i < a_{i+1} < \dots < a_j$, $0 \leq i < j < n$) i proceni mu složenost.

Problem (najveći težinski zbir rotacija niza): Dat je niz a celih brojeva dužine n . Dozvoljena je operacija cikličnog pomeranja tj. rotacije niza ulevo za proizvoljan broj mesta. Napisati program kojim se određuje najmanji broj pomeranja ulevo tako da težinski zbir

$$\sum_{i=0}^{n-1} i \cdot a_i = 0 \cdot a_0 + 1 \cdot a_1 + 2 \cdot a_2 + 3 \cdot a_3 + \dots + (n-1) \cdot a_{n-1}$$

u transformisanom nizu ima najveću vrednost.

Problem (postavljanje rutera): Duž jedne ulice su ravnomerno raspoređene zgrade (rastojanje između svake dve susedne je jednako). Za svaku

zgradu je poznat broj korisnika koje novi dobavljač interneta treba da poveže. Odrediti u koju od zgrada treba postaviti ruter tako da bi ukupna dužina optičkih kablova kojim se svaki od korisnika povezuje sa ruterom bila minimalna (računati samo dužinu kablova od zgrade do zgrade i zanemariti dužine unutar zgrada). Na primer, ako je broj korisnika po zgradama jednak 3, 5, 1, 6, 2, 4, ruter treba postaviti u četvrtu zgradu sleva i dužina kablova je tada jednaka $3 \cdot 3 + 2 \cdot 5 + 1 \cdot 1 + 1 \cdot 2 + 2 \cdot 4 = 30$.

Problem (tačna perioda niza): Dat je niz a dužine 2^k . Broj p je tačna perioda niza a ako se niz a može dobiti ponavljanjem podniza prvih p elemenata niza, bez dodatnih elemenata. Na primer, dužina najkraće tačne periode niza 1 2 3 4 1 2 3 4 je 4, dok je za niz 1 2 3 1 2 3 1 2 dužina najkraće tačne periode 8). Definirati funkciju koja efikasno određuje dužinu p najmanje tačne periode tog niza i proceni joj složenost.

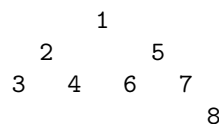
Problem (provera duplikata u nizu): Definirati funkciju koja efikasno proverava da li u datom nizu celih brojeva ima duplikata i proceni joj složenost.

Problem (duplikati u nizu od 0 do $n - 1$): Dat je niz od n brojeva u kom su svi elementi iz intervala od 0 do $n - 1$. Neki možda nedostaju, a neki se ponavljaju. Napisati program koji ispisuje sve one koji nedostaju i sve one koji se ponavljaju (u proizvoljnom redosledu).

Čas 3.1, 3.2, 3.3

Problem (Grejov kod): Grejov kôd dužine 2^n definiše se kao niz koji sadrži sve zapise n -tocifrenih binarnih brojeva takve da se svaka dva susedna zapisa (kao i prvi i poslednji zapis) razlikuju tačno u jednom bitu. Na primer 00, 01, 11, 10 je Grejov kôd dužine 2^2 . Zaista 00 i 01 se razlikuju samo na drugom bitu, 01 i 11 samo na prvom, 11 i 10 samo na drugom, a 10 i 00 samo na prvom bitu. Definirati funkciju koja konstruiše Grejov kôd dužine 2^n za proizvoljno n .

Problem (određivanje prefiksnog na osnovu postfiksnog i infiksnog obilaska binarnog drveta): Poznati su infiksni i postfiksni obilazak drveta (čvorovi drveta sadrže karaktere i obilasci su zadati pomoću dve niske iste dužine). Napisati program koji na osnovu njih određuje prefiksni obilazak. Pretpostaviti da svi čvorovi u drvetu sadrže različite vrednosti (tada je rešenje jedinstveno). Na primer, razmotrimo naredno drvo.



Njegov postfiksni obilazak je 34268751 a infiksni 32416578. Potrebno je da rekonstruišemo prefiksni obilazak koji je u ovom slučaju 12345678.

Problem (rotiranje elemenata niza za k mesta ulevo): Neka je dat niz od n elemenata. Definirati funkciju složenosti $O(n)$ koja njegov sadržaj rotira za k mesta ulevo, bez korišćenja pomoćnog niza.

Problem (Morzeov niz): Niz

$$1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, \dots,$$

koji se sastoji od nula i jedinica, gradi se na sledeći način: prvi element je 1; drugi se dobija logičkom negacijom prvog $!1 = 0$, treći i četvrti logičkom negacijom prethodna dva $!1 = 0, !0 = 1$, peti, šesti, sedmi i osmi logičkom negacijom prva četiri - dobija se 0, 1, 1, 0 itd. Dakle, krenuvši od jednočlanog segmenta 1, svakom početnom segmentu koji je dužine 2^k (k uzima vrednosti $0, 1, 2, \dots$) dopisuje se segment iste dužine dobijen logičkom negacijom svih elemenata početnog segmenta. Definirati funkciju koja za dato n određuje n -ti član niza (brojanje kreće od 1).

Problem (zvezda): U nekoj grupi ljudi osoba se naziva zvezda (engl. superstar) ako je svi prisutni znaju, a ona ne poznaje nikoga od prisutnih. Definirati funkciju za ispitivanje da li u datom skupu ljudi postoji zvezda. Data je matrica kojom se određuje ko koga poznaje.

Problem (apsolutni pobednik na glasanju): Održano je glasanje i glasalo se za više kandidata. Osoba je apsolutni pobednik ako je dobila strogo više glasova nego svi ostali kandidati zajedno. Definirati algoritam koji na osnovu niza svih glasačkih listića sa glasanja određuje da li postoji apsolutni pobednik i koji je.

Problem (Hornerova shema): Dat je niz realnih brojeva $a_0, a_1, \dots, a_n$ i realni broj x . Definirati bar dva algoritma koji određuju vrednost polinoma $P_n(x) = \sum_{i=0}^n a_i x^i$ korišćenjem najviše $O(n)$ množenja i $O(n)$ sabiranja.

Problem (faktori ravnoteže binarnog drveta): Visina nepraznog binarnog drveta definiše se kao rastojanje od korena do njegovog najdaljeg potomka, ako takav potomak postoji, tj. nula ako koren nema potomaka. Visina praznog drveta je nula. Faktor ravnoteže čvora v binarnog drveta definiše se kao razlika visina njegovog levog i desnog poddrveta. Definirati funkciju koja za svaki čvor drveta izračunava faktor ravnoteže.

Problem (dijametar binarnog drveta): Rastojanje između dva čvora binarnog drveta je broj grana na jedinstvenom putu koji ih povezuje. Dijametar drveta je najveće moguće rastojanje dva njegova čvora. Konstruisati efikasan algoritam za određivanje dijametra datog drveta i odrediti njegovu vremensku složenost.

Problem (maksimalni zbir segmenta - Kadanov algoritam): Definirati efikasanu funkciju koja pronalazi najveći mogući zbir segmenta (podniza uzastopnih elemenata) datog niza brojeva. Proceni joj složenost.

Problem (maksimalni zbir segmenta - zbrovi prefiksa): Definirati

efikasanu funkciju koja pronalazi najveći mogući zbir segmenta (podniza uzastopnih elemenata) datog niza brojeva. Proceni joj složenost.

Problem (maksimalni zbir segmenta deljiv sa k): Dat je niz od n nenegativnih celih brojeva. Definirati efikasan algoritam koji određuje najveći zbir koji ima neki njegov segment (podniz uzastopnih elemenata) koji je deljiv datim brojem k .

Problem (broj rastućih segmenata - linearna složenost): Dat je niz a celih brojeva. Definirati algoritam složenosti $O(n)$ kojim se određuje koliko u tom nizu postoji rastućih segmenata (rastući segment čine uzastopni elementi niza $a_i < a_{i+1} < \dots < a_j$, $0 \leq i < j < n$) i proceni mu složenost.

Problem: Napiši program koji određuje najveći zbir podniza datog niza nenegativnih brojeva koji ne sadrži dva uzastopna člana niza. Na primer, za niz 7, 3, 2, 4, 1, 5 najveći takav podniz je 7, 4, 5, čiji je zbir 16.

Čas 4.1, 4.2, 4.3

Problem: Pregledač veba pamti istoriju posećenih sajtova i korisnik ima mogućnost da se vraća unatrag na sajtove koje je ranije posetio. Napisati program koji simulira istoriju pregledača tako što učitavaju adrese posećenih sajtova (svaka u posebnom redu), a kada se učita red u kome piše **back** pregledač se vraća na poslednju posećenu stranicu.

Problem: Napisati program koji sve učitane linije sa standardnog ulaza ispisuje u obratnom redosledu (upotrebiti stek).

Problem: Napisati program koji proverava da li su zagrade u infiksno zapisanom izrazu korektno uparene. Moguće je pojavljivanje malih, srednjih i velikih zagrada (tzv. običnih, uglastih i vitičastih).

Problem: Napisati program koji izračunava vrednost ispravnog postfiksno zadatog izraza koji sadrži samo jednocifrene brojeve i operatore $+$ i $*$. Na primer, za izraz $34+5*$ program treba da izračuna vrednost 35.

Problem: Dat je ispravan infiksni aritmetički izraz koji ima zagrade oko svake primene binarnog operatora. Napisati program koji ga prevodi u postfiksni oblik. Na primer, $((3*5)+(7+(2*1)))*4$ se prevodi u $35*721*++4*$. Jednostavnosti radi pretpostaviti da su svi operandi jednocifreni brojevi i da se javljaju samo operacije sabiranja i množenja.

Problem: Dat je ispravan infiksni aritmetički izraz koji ima zagrade oko svake primene binarnog operatora. Na primer, $((3*5)+(7+(2*1)))*4$ Napisati program koji izračunava njegovu vrednost (za izraz u zagradama to je vrednost 96). Jednostavnosti radi pretpostaviti da su svi operandi jednocifreni brojevi i da se javljaju samo operacije množenja i sabiranja.

Problem: Napisati program koji vrši prevođenje ispravnog infiksno zadatog izraza u njemu ekvivalentan prefiksno zadati izraz. Pretpostaviti da su svi brojevi jednocifreni. Na primer, za izraz $2*3+4*(5+6)$ program treba da ispiše $23*456+**$.

Problem: Napisati program koji izračunava vrednost ispravno zadatog infiksno zapisanog izraza koji sadrži operatore $+$ i $*$. Jednostavnosti radi pretpostaviti da su svi operandi jednocifreni brojevi. Na primer, za izraz $(3+4)*5+6$ program treba da ispiše 41 .

Problem: Za svaki element niza neoznačenih brojeva odrediti njemu najbliži prethodnik u nizu koji je manji od njega (ako takav element ne postoji, ispisati $-$). Na primer, za niz $3, 7, 4, 2, 6, 5$, ispisati $-, 3, 3, -, 2, 2$.