

Ocena efikasnosti algoritma - Zadaci

Istorija revizija

<i>Verzija</i>	<i>Autori</i>	<i>Datum</i>	<i>Status</i>	<i>Komentar</i>
1.00w	Saša Tošić	16 - Mar - 2018	Inicijalna verzija	Početna verzija dokumenta
1.1	Marko Vasiljević	15-Apr-2018	Radna verzija	Prilagođen online kursu Cosnovni

Reference

Sadržaj

Zadatak 1:	Broj nula cifara	5
Zadatak 2:	Kvadratna jednačina	5
Zadatak 3:	Koren broja	5
Zadatak 4:	Broj ocena 2.....	5
Zadatak 5:	Broj ocena 1.....	5
Zadatak 6:	Podniz sa najvećim zbirom 3	5
Zadatak 7:	Podniz sa najvećim zbirom 2	6
Zadatak 8:	Podniz sa najvećim zbirom 1	6

Zadatak 1: Broj nula cifara

Napisati program koji učitava pridodan broj n i ispisuje koliko u broju n postoji cifara 0. Algoritam realizovati u efikasnosti $O(\log_{10}(n))$.

Ideja: odvajati jednu po jednu cifru broja n i proveravati da li je dobijena cifra 0.

Zadatak 2: Kvadratna jednačina

Pronaći približno rešenje kvadratne jednačine $n^2 - 5 \cdot n + 2$ na intervalu $[1, 10]$. Algoritam realizovati u efikasnosti $O(\log_2(n))$.

Ideja: za vrednost 1, jednačina daje vrednost manju od 0, a za vrednost 10 veću od 0. Deliti interval na pola traženjem sredine intervala, i ako se dobije broj manji od 0, tada odbaciti levu polovinu intervala, a u suprotnom odbaciti desnu polovinu intervala.

Zadatak 3: Koren broja

Napisati program koji učitava broj n veći od 1 i pronalazi i ispisuje koren broja n , na dve decimale. Algoritam realizovati u efikasnosti $O(100 \cdot \sqrt{n})$.

Ideja: krenuti od broja 1 i dodavati mu 0.01, sve dok se ne dobije vrednost n takva da je $n^2 \leq n < (n + 0.01)^2$. Ispisati dobivenu vrednost x .

Zadatak 4: Broj ocena 2

Neka je dat niz od n školskih ocena iz intervala $[1, 5]$. Ispisati kojih ocena ima najviše. Algoritam rešiti u efikasnosti $O(n \cdot k)$ gde je k najveća moguća ocena, u ovom slučaju ocena 5.

Ideja: Za svaku ocenu od 1 do 5, prebrojati koliko ima tih ocena, i videti kojih ocena ima najviše.

Zadatak 5: Broj ocena 1

Neka je dat niz od n školskih ocena iz intervala $[1, 5]$. Ispisati kojih ocena ima najviše. Algoritam rešiti u efikasnosti $O(n)$.

Ideja: Napraviti niz brojača (za svaku ocenu po jedan brojač). Proći kroz sve ocene i povećati brojač za datu ocenu za 1. Nakon toga, pronaći koji je brojač najveći.

Zadatak 6: Podniz sa najvećim zbirom 3

Napisati program koji učitava članove niza a od n elemenata i unutar njega pronalazi i ispisuje podniz uzastopnih elemenata datog niza koji ima najveći mogući zbir. Algoritam realizovati u efikasnosti $O(n^3)$.

Primer: neka je dat niz 2, -4, 3, 8, 4, -2, 3, -6, 2, 1 sa 10 elemenata. Tada je traženi podniz niz brojeva 3, 8, 4, -2 i 3 i ima zbir 16.

Ideja: Sa dve **for** petlje generisati granice za sve podnizove, pri čemu prva petlja definiše početak niza, a druga kraj niza. Za svaki generisani podniz, sa novom **for** petljom izračunati sumu elemenata tog podniza i ukoliko se dobije veća suma, zapamtiti granice podniza. Program u sebi ima 3 ugnježdene **for** petlje.

Zadatak 7: Podniz sa najvećim zbirom 2

Napisati program koji učitava članove niza **a** od **n** elemenata i unutar njega pronalazi i ispisuje podniz uzastopnih elemenata datog niza koji ima najveći mogući zbir. Algoritam realizovati u efiksnosti $O(n^2)$.

Primer: neka je dat niz 2, -4, **3, 8, 4**, -2, 3, -6, 2, 1 sa 10 elemenata. Tada je traženi podniz niz brojeva 3, 8, 4, -2 i 3 i ima zbir 16.

Ideja: Kao pripremu za rad, kreirati niz **c**, takav da je **c[i]** suma svih elemenata niza **a**, od nultog do **i**-tog elementa. Za ovu pripremu dovoljna je jedna **for** petlja i formula da je **c[i] = a[i] + c[i - 1]**, pri čemu je **c[0] = a[0]**. Nakon toga, sa dve **for** petlje generisati granice za sve podnizove, pri čemu prva petlja definiše početak niza, a druga kraj niza. Za svaki generisani podniz, pomoću niza **c** izračunati zbir elemenata u podnizu i ukoliko se dobije veća suma, zapamtiti granice podniza. Zbir elemenata od **i**-tog do **j**-tog elementa se definiše kao **c[j] - c[i - 1]**. Program u sebi ima 2 ugnježdene **for** petlje, a **for** petlja za računanje niza **c** se zanemaruje u efikasnosti algoritma.

Zadatak 8: Podniz sa najvećim zbirom 1

Napisati program koji učitava članove niza **a** od **n** elemenata i unutar njega pronalazi i ispisuje podniz uzastopnih elemenata datog niza koji ima najveći mogući zbir. Algoritam realizovati u efiksnosti $O(n)$.

Primer: neka je dat niz 2, -4, **3, 8, 4**, -2, 3, -6, 2, 1 sa 10 elemenata. Tada je traženi podniz niz brojeva 3, 8, 4, -2 i 3 i ima zbir 16.

Ideja: sabirati elemente od početka niza i pamtit granice podniza sa najvećim zbirom, kao i njegov zbir. Svaki put kada zbir elemenata bude negativan, zanemariti do sada izračunati zbir i krenuti sabiranjem iz početka. Na primer, u datom primeru, prvo se kreće od broja 2 i to je najbolji zbir za sada. Kada se doda broj -4, zbir postaje -2, i tada zbir postavimo na 0 i nastavljamo od broja 3. Kako je 3 veće od 2, pamtimo da je to bolji zbir. Algoritam tako nastavlja da funkcioniše i za preostale elemente datog niza.