

Metode sortiranja nizova

Sortiranje niza u neopadajućem ili nerastućem poretku podrazumeva nalaženje jedne permutacije elemenata niza u kojoj se elementi pojavljuju u neopadajućem tj. nerastućem poretku.

Selection sort

Metoda sortiranja izborom najmanjeg elementa odnosi se na sortiranje niza podataka x sa n elemenata u neopadajući poredak (slično izbor najvećeg elementa obezbeđuje sortiranje u nerastuci poredak). Prvo se nalazi najmanji element niza i on se "dovodi" na prvo mesto, zatim se nalazi najmanji od preostalih $n - 1$ elemenata i on se "dovodi" na drugo mesto, nalazi najmanji od preostalih $n-2$ elemenata i dovodi na treće mesto, itd, zaključno sa nalaženjem manjem od poslednja dva elementa i njegovim "dovođenjem" na pretposlednje mesto. Na poslednjem mestu će ostati element koji nije manji ni od jednog u nizu (najveći element).

Implementacije algoritama sortiranja

```
/* Poredimo i-ti i j-ti element niza a */
```

```
int uporedi(int a[], int i, int j) {  
    br_poredjenja++;  
    return a[i] > a[j];  
}
```

```
/* Menjamo i-ti i j-ti element niza a */
```

```
void zameni(int a[], int i, int j) {  
    br_zamena++;  
    int tmp = a[i];  
    a[i] = a[j];  
    a[j] = tmp;  
}
```

```
/* Selection sort */
```

```
void selection_sort_1(int a[], int n) {  
    int i, j;  
    for (i = 0; i < n-1; i++)  
        for (j = i+1; j < n; j++)  
            if (uporedi(a, i, j))  
                zameni(a, i, j);  
}
```

```
void selection_sort_2(int a[], int n) {  
    int i, j, min;  
    for (i = 0; i < n - 1; i++) {  
        min = i;  
        for (j = i+1; j < n; j++)  
            if (uporedi(a, min, j))  
                min = j;  
        if (i != min)  
            zameni(a, i, min);  
    }  
}
```

Šta možete reći o broju izvršenih zamena i poređenja kod ove dve implementacije? Vidi eksperiment dole!

Insertion sort

Ako je dat niz (x_n) sa elementima nekog, uređenog tipa T, koji treba urediti u neopadajući poredak, ova metoda sortiranja polazi od pretpostavke da imamo uređen početni deo niza, (to svakako važi za $i = 2$, jer je podniz sa jednim elementom uređen) i u svakom koraku, počevši od $i = 2$ i povećanjem i , i -ti element se stavlja na pravo mesto u odnosu na prvih (uređenih) $i - 1$.

/* Insertion sort */

```
void insertion_sort_1(int a[], int n) {
    int i, j;
    for (i = 1; i < n; i++)
        for (j = i; j > 0 && uporedi(a, j-1, j); j--)
            zameni(a, j-1, j);
}
```

/* Broj dodela se moze redukovati tako sto se umesto zamena koristi dodela privremenoj promenljivoj ****/***

```
void insertion_sort_2(int a[], int n) {
    int i, j;
    for (i = 1; i < n; i++) {
        int tmp = a[i];
        for (j = i; j > 0 && a[j-1] > tmp; j--){
            a[j] = a[j-1];
        }
        a[j] = tmp;
    }
}
```

Šta možete reći o broju izvršenih zamena i poređenja kod ove dve implementacije? Vidi eksperiment dole!

Bubble sort

Ova metoda je elementarna: ponavlja se prolazak kroz niz elemenata i razmenjuju se susedni elementi, ako je potrebno, sve dok se ne završi prolaz koji ne zahteva nijednu razmenu. Tada je niz sortiran.

/* Bubble Sort */

/* Najjednostavnija implementacija - menjamo uzastopne dok god ima promena */

```
void bubble_sort_1(int a[], int n) {
    int bilo_zamena;
    do {
        int i;
        bilo_zamena = 0;
        for (i = 0; i < n-1; i++)
            if (uporedi(a, i, i+1)) {
                zameni(a, i, i+1);
                bilo_zamena = 1;
            }
    } while(bilo_zamena);
}
```

/* Malo ubrzanje se moze postici ukoliko se primeti da posle svakog prolaza najveći ispliva na kraj, tako da se svaki naredni prolaz skracuje za jednu poziciju. ****/***

```
void bubble_sort_2(int a[], int n) {
```

```

int i, j;
for (i = n-1; i > 0; i--)
    for (j = 0; j < i; j++)
        if (uporedi(a, j, j+1))
            zameni(a, j, j+1);
}

/*    Kombinujemo prethodnu optimizaciju sa
    prvobitnim kriterijumom zaustavljanja. */
void bubble_sort_3(int a[], int n) {
    int i, j;
    int bilo_zamena = 1;
    for(i = n-1; bilo_zamena; i--) {
        bilo_zamena = 0;
        for (j = 0; j < i; j++) {
            if (uporedi(a, j, j+1)) {
                zameni(a, j, j+1);
                bilo_zamena = 1;
            }
        }
    }
}

```

Šta možete reći o broju izvršenih zamena i poređenja kod ove tri implementacije? Vidi eksperiment dole!

Shell sort

Šelsort je jednostavno proširenje sortiranja umetanjem koje dopušta direktnu razmenu udaljenih elemenata. Proširenje se sastoji u tome da se kroz algoritam umetanja prolazi više puta, u prvom prolazu, umesto koraka 1 uzima se neki korak h koji je manji od n (što omogućuje razmenu udaljenih elemenata) i tako se dobija h -sortiran niz, tj. niz u kome su elementi na rastojanju h sortirani, mada susedni elementi to ne moraju biti. U drugom prolazu kroz isti algoritam sprovodi se isti postupak ali za manji korak h . Sa prolazima se nastavlja sve do koraka $h = 1$, u kome se dobija potpuno sortirani niz.

```

/*Shell sort*/
void shell_sort_1(int a[], int n) {
    int sirina;
    int i, j, k;
    for (sirina = n/2; sirina >= 1; sirina /= 2)
        for (k = 0; k < sirina; k++)
            /* Elementi k-te kolone su:
               a[k], a[k+sirina], a[k+2*sirina],...
               Sortiramo je koristeći insertion sort...
            */
            for (i = k + sirina; i < n; i += sirina)
                for (j = i; j > k && uporedi(a, j-sirina, j); j -= sirina)
                    zameni(a, j-sirina, j);
}

/*    Malo pametnija implementacija - ne sortiramo jednu kolonu nakon druge,
    vec dopustamo da sortiranje tece paralelno. Nakon sto element neke kolone
    pronadje svoje mesto, prelazimo na obradu elementa do njega,
    umesto elementa ispod njega. Ovim se ne dobija sustinsko ubrzanje,
    ali cini kod malo jednostavnijim. */

void shell_sort_2(int a[], int n) {
    int sirina;
    int i, j, k;
    for (sirina = n/2; sirina >= 1; sirina /= 2)

```

```

    for (i = sirina; i < n; i++)
        for (j = i; j >= sirina && uporedi(a, j-sirina, j); j -= sirina)
            zameni(a, j-sirina, j);
}

```

Quick sort

Ovo je najčešće upotrebljavan algoritam sortiranja. Osnovni oblik algoritma dao je 1960, Hor (Hoare). Nije težak za implementaciju, a koristi manje resursa (vremena i prostora) nego bilo koji drugi algoritam sortiranja, u većini slučajeva. Algoritam ne zahteva dodatnu memoriju, samo $n \cdot \log(n)$ operacija u proseku za sortiranje n elemenata, i ima izuzetno kratku unutrašnju petlju. Loše strane algoritma su što je rekurzivan (nerekurzivna varijanta je mnogo složenija), u najgorem slučaju izvršava oko n^2 operacija. Postoje i verzije ovog algoritma koje ga poboljšavaju. Algoritam je vrlo osetljiv na implementaciju (efikasnost se može narušiti lošim izborom npr. pivota u implementaciji). Ako se ne želi analizirati najbolja implementacija, bolje je primeniti šelsort. Ideja algoritma sastoji se u particioniranju niza prema odabranom elementu particioniranja koji se dovodi na pravo mesto, i u primeni algoritma brzog sortiranja na svaku od dve dobijene particije. Rekurzivni poziv se završava kada se primeni na particiju sa manje od dva elementa.

/* Quick sort */

```

void quicksort(int a[], int l, int d) {
    int s = (l+d) / 2;
    int piv = a[s], t, m;
    if (l >= d)
        return;
    /* Pvi korak je razdvojiti niz tako da bude oblika
    < < < < piv >= >= >=
    Ovo se odvija u nekoliko koraka. */
    swap(a, l, s);

```

```

    /* piv x x x x x x x x */
    /* piv < < < < >= >= >= x x */
    /*      m      t */

```

```

    m = l;
    for (t = l+1; t <= d; t++)
        if (a[t] < piv)
            swap(a, t, ++m);

```

```

    /* piv < < < < >= >= >= >= */
    /*      m      */

```

```

    swap(a, l, m);

```

```

    /* < < < < piv >= >= >= >= */

```

```

    quicksort(a, l, m-1);
    quicksort(a, m+1, d);
}

```

Merge sort

Sortiranje spajanjem ili "merge sort" je algoritam sortiranja zasnovan na poređenju. To je rekurzivni algoritam. Njegova vremenska složenost proporcionalna je sa $O(n \cdot \log(n))$, a u srednjem slučaju je uvek efikasniji od algoritma brzog sortiranja (quick sort). U većini implementacija je stabilan, što znači da zadržava početni redosled jednakih elemenata u sortiranom nizu. Predstavlja primer algoritamske paradigme "podeli pa vladaj". Konstruisao ga je Džon fon Nojman (John von Neumann) 1945. godine. Konceptualno, algoritam sortiranja spajanjem "radi" na sledeći način:

- 1. Ako niz ima nula ili jedan element, onda je vec soritran. Inace,
 - 2. Podeliti nesortirani niz u dva podniza priblizno jednake duzine.
 - 3. Sortirati svaki podniz rekurzivno ponovnom primenom algoritma sortiranja spajanjem.
 - 4. Spojiti dva sortirana podniza u jedan sortirani niz.
- Algoritam sortiranja spajanjem ukljucuje dva vazna principa kojima poboljsava (smanjuje) vreme izvršavanja:
- 1. kratki niz je moguće sortirati u manjem broju koraka nego dugacki (osnova za deljenje niza na dva podniza)
 - 2. manje koraka je potrebno za konstrukciju sortiranog niza od dva sortirana podniza nego od dva nesortirana podniza (osnova za spajanje).

```
/* Merge sort */
void mergesort(int a[], int l, int d) {
    int s = (l + d)/2;
    static int b[100];
    int i, j, k;
    if (l >= d)
        return;
    mergesort(a, l, s);
    mergesort(a, s+1, d);
    i = l;
    j = s+1;
    k = 0;
    while (i <= s && j <= d) {
        if (a[i] < a[j]) {
            b[k++] = a[i++];
        }
        else {
            b[k++] = a[j++];
        }
    }
    while(i <= s)
        b[k++] = a[i++];
    while(j <= d)
        b[k++] = a[j++];

    for (k = 0, i = l; i<=d; k++, i++)
        a[i] = b[k];
}
```

Složenost algoritma (vremenska ili prostorna) je obično **neka funkcija** koja povezuje veličinu problema (ulaza) sa brojem koraka izvršavanja algoritma(**vremenska složenost**) ili brojem potrebnih memorijskih lokacija (**prostorna složenost**). Uobičajeno je da se složenost algoritama (vremenska ili prostorna) procenjuje u asimptotskom smislu, tj. da se funkcija složenosti procenjuje za dosta velike dužine ulaza. Za to se koriste "veliko O" notacija(**O()**), "omega notacija" (**Ω()**), i "tetanotacija" (**Θ()**). "Veliko O" notacija, poznata kao Landau ili Bahman-Landau notacija, **opisuje granično ponašanje funkcije** kada argument teži nekoj specifičnoj vrednosti ili beskonačnosti, obično u terminima jednostavnijih funkcija. "Veliko O" notacija - dobila je ime od "order of" ili "red veličine" pa ćemo za vremensku složenost koja je reda $O(f(n))$ govoriti da je "proporcionalna" sa $f(n)$. Analiza vremenske složenosti algoritma **ne mora biti precizna** (da prebroji svaki korak u izvršavanju algoritma), već je dovoljno da odredi najveće elemente takvih proračuna.

Analiza složenosti			
Metoda sortiranja	Vremenska složenost	Prostorna složenost	Stabilnost
Selection sort	$O(n^2)$	$O(1)$	Stabilan
Insertion sort	$O(n^2)$	$O(1)$	Stabilan

Bubble sort	$O(n^2)$	$O(1)$	Stabilan
Shell sort	$O(n\log^2(n))$	$O(1)$	Zavisi
Merge sort	$O(n\log(n))$	$O(n)$	Stabilan
Quicksort	$O(n\log(n))$	$O(\log(n))$	Zavisi

Eksperimentalni test elementarnih metoda sortiranja

Svi nizovi su nizovi slučajnih generisanih brojeva.

Dimenzija niza: 1000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	252215	499500	0.00
Selection sort 2	992	499500	0.00
Insertion sort 1	255988	256980	0.01
Insertion sort 2	171324	255988	0.00
Bubble sort 1	255988	971028	0.01
Bubble sort 2	255988	499500	0.01
Bubble sort 3	255988	499122	0.01
Shell sort 1	7800	15269	0.00
Shell sort 2	7800	15269	0.00
Dimenzija niza: 2000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	983804	1999000	0.02
Selection sort 2	1991	1999000	0.02
Insertion sort 1	997506	999502	0.01
Insertion sort 2	666336	997506	0.01
Bubble sort 1	997506	3910044	0.05

Bubble sort 2	997506	1999000	0.03
Bubble sort 3	997506	1998054	0.02
Shell sort 1	19987	36950	0.01
Shell sort 2	19987	36950	0.00
Dimenzija niza: 3000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	2177552	4498500	0.05
Selection sort 2	2988	4498500	0.04
Insertion sort 1	2239865	2242859	0.03
Insertion sort 2	1495242	2239865	0.02
Bubble sort 1	2239865	8757080	0.10
Bubble sort 2	2239865	4498500	0.06
Bubble sort 3	2239865	4495340	0.07
Shell sort 1	30905	59357	0.00
Shell sort 2	30905	59357	0.00
Dimenzija niza: 4000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	3834467	7998000	0.11
Selection sort 2	3985	7998000	0.06
Insertion sort 1	3991969	3995956	0.08
Insertion sort 2	2663978	3991969	0.02
Bubble sort 1	3991969	15672081	0.20
Bubble sort 2	3991969	7998000	0.10
Bubble sort 3	3991969	7994760	0.12
Shell sort 1	48205	86240	0.00
Shell sort 2	48205	86240	0.00
Dimenzija niza: 5000 elemenata			
Algoritam	Zamena	Poređenja	Vreme

Selection sort 1	5816139	12497500	0.16
Selection sort 2	4994	12497500	0.09
Insertion sort 1	6173014	6178008	0.12
Insertion sort 2	4118675	6173014	0.04
Bubble sort 1	6173014	24470105	0.29
Bubble sort 2	6173014	12497500	0.17
Bubble sort 3	6173014	12492040	0.19
Shell sort 1	62172	114645	0.00
Shell sort 2	62172	114645	0.00
Dimenzija niza: 6000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	8387268	17997000	0.22
Selection sort 2	5994	17997000	0.12
Insertion sort 1	8935659	8941652	0.16
Insertion sort 2	5961105	8935659	0.06
Bubble sort 1	8935659	34758206	0.44
Bubble sort 2	8935659	17997000	0.25
Bubble sort 3	8935659	17975885	0.25
Shell sort 1	71831	134821	0.00
Shell sort 2	71831	134821	0.00
Dimenzija niza: 7000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	11549379	24496500	0.33
Selection sort 2	6985	24496500	0.16
Insertion sort 1	12419996	12426983	0.19
Insertion sort 2	8284663	12419996	0.10
Bubble sort 1	12419996	48419082	0.60
Bubble sort 2	12419996	24496500	0.34
Bubble sort 3	12419996	24493179	0.34

Shell sort 1	91274	164592	0.01
Shell sort 2	91274	164592	0.00
Dimenzija niza: 8000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	15034475	31996000	0.40
Selection sort 2	7990	31996000	0.23
Insertion sort 1	16358048	16366040	0.26
Insertion sort 2	10910698	16358048	0.12
Bubble sort 1	16358048	63544056	0.76
Bubble sort 2	16358048	31996000	0.48
Bubble sort 3	16358048	31994460	0.46
Shell sort 1	118159	202043	0.00
Shell sort 2	118159	202043	0.00
Dimenzija niza: 9000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	18418713	40495500	0.52
Selection sort 2	8989	40495500	0.28
Insertion sort 1	20381206	20390199	0.36
Insertion sort 2	13593470	20381206	0.16
Bubble sort 1	20381206	80019108	0.94
Bubble sort 2	20381206	40495500	0.61
Bubble sort 3	20381206	40489722	0.61
Shell sort 1	116932	220243	0.00
Shell sort 2	116932	220243	0.01
Dimenzija niza: 10000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	22244658	49995000	0.62
Selection sort 2	9991	49995000	0.35

Insertion sort 1	24836048	24846033	0.43
Insertion sort 2	16564031	24836048	0.19
Bubble sort 1	24836048	98920107	1.19
Bubble sort 2	24836048	49995000	0.68
Bubble sort 3	24836048	49989329	0.73
Shell sort 1	143988	259003	0.01
Shell sort 2	143988	259003	0.00
Dimenzija niza: 11000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	26820118	60494500	0.73
Selection sort 2	10993	60494500	0.44
Insertion sort 1	30086108	30097098	0.51
Insertion sort 2	20064738	30086108	0.24
Bubble sort 1	30086108	119361148	1.44
Bubble sort 2	30086108	60494500	0.86
Bubble sort 3	30086108	60483622	0.84
Shell sort 1	154895	281164	0.01
Shell sort 2	154895	281164	0.01
Dimenzija niza: 12000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	31659507	71994000	0.91
Selection sort 2	11992	71994000	0.51
Insertion sort 1	36225469	36237465	0.62
Insertion sort 2	24158312	36225469	0.28
Bubble sort 1	36225469	142740104	1.71
Bubble sort 2	36225469	71994000	1.33
Bubble sort 3	36225469	71988644	1.17
Shell sort 1	176985	314834	0.00
Shell sort 2	176985	314834	0.01

Dimenzija niza: 13000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	36841987	84493500	1.04
Selection sort 2	12987	84493500	0.60
Insertion sort 1	42244155	42257146	0.72
Insertion sort 2	28171436	42244155	0.32
Bubble sort 1	42244155	167284131	1.96
Bubble sort 2	42244155	84493500	1.21
Bubble sort 3	42244155	84484985	1.22
Shell sort 1	206109	355406	0.00
Shell sort 2	206109	355406	0.01
Dimenzija niza: 14000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	41912107	97993000	1.25
Selection sort 2	13987	97993000	0.67
Insertion sort 1	48835662	48849655	0.84
Insertion sort 2	32566440	48835662	0.37
Bubble sort 1	48835662	194978072	2.34
Bubble sort 2	48835662	97993000	1.37
Bubble sort 3	48835662	97990444	1.47
Shell sort 1	219822	380774	0.01
Shell sort 2	219822	380774	0.01
Dimenzija niza: 15000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	48278918	112492500	1.40
Selection sort 2	14993	112492500	0.75
Insertion sort 1	56462595	56477583	1.00
Insertion sort 2	37651729	56462595	0.43

Bubble sort 1	56462595	223305112	2.68
Bubble sort 2	56462595	112492500	1.63
Bubble sort 3	56462595	112486284	1.59
Shell sort 1	225782	397938	0.01
Shell sort 2	225782	397938	0.01
Dimenzija niza: 16000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	54047647	127992000	1.56
Selection sort 2	15990	127992000	0.86
Insertion sort 1	64253587	64269579	1.10
Insertion sort 2	42846390	64253587	0.51
Bubble sort 1	64253587	253760139	2.92
Bubble sort 2	64253587	127992000	1.76
Bubble sort 3	64253587	127982409	1.76
Shell sort 1	299012	482954	0.01
Shell sort 2	299012	482954	0.00
Dimenzija niza: 17000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	60403631	144491500	1.71
Selection sort 2	16991	144491500	0.98
Insertion sort 1	72282331	72299320	1.15
Insertion sort 2	48199553	72282331	0.52
Bubble sort 1	72282331	285566201	3.33
Bubble sort 2	72282331	144491500	2.04
Bubble sort 3	72282331	144471400	2.07
Shell sort 1	278302	490483	0.01
Shell sort 2	278302	490483	0.01
Dimenzija niza: 18000 elemenata			

Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	67420800	161991000	1.99
Selection sort 2	17988	161991000	1.13
Insertion sort 1	81284316	81302300	1.41
Insertion sort 2	54201543	81284316	0.63
Bubble sort 1	81284316	318492305	3.80
Bubble sort 2	81284316	161991000	2.31
Bubble sort 3	81284316	161944640	2.27
Shell sort 1	295603	520405	0.01
Shell sort 2	295603	520405	0.01
Dimenzija niza: 19000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	74242037	180490500	2.16
Selection sort 2	18987	180490500	1.25
Insertion sort 1	90513664	90532659	1.51
Insertion sort 2	60355108	90513664	0.66
Bubble sort 1	90513664	353419398	4.19
Bubble sort 2	90513664	180490500	2.54
Bubble sort 3	90513664	180411497	2.54
Shell sort 1	302057	539127	0.01
Shell sort 2	302057	539127	0.00
Dimenzija niza: 20000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	80778193	199990000	2.43
Selection sort 2	19990	199990000	1.42
Insertion sort 1	100240087	100260080	1.69
Insertion sort 2	66840057	100240087	0.76
Bubble sort 1	100240087	395780210	4.68
Bubble sort 2	100240087	199990000	2.83

Bubble sort 3	100240087	199968055	2.82
Shell sort 1	357487	607407	0.01
Shell sort 2	357487	607407	0.01
Dimenzija niza: 21000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	87735545	220489500	2.65
Selection sort 2	20986	220489500	1.56
Insertion sort 1	109428732	109449718	1.88
Insertion sort 2	72966487	109428732	0.83
Bubble sort 1	109428732	437871148	5.31
Bubble sort 2	109428732	220489500	3.14
Bubble sort 3	109428732	220478622	3.17
Shell sort 1	427635	689918	0.02
Shell sort 2	427635	689918	0.01
Dimenzija niza: 22000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	95904561	241989000	2.96
Selection sort 2	21992	241989000	1.69
Insertion sort 1	120349649	120371636	2.10
Insertion sort 2	80247765	120349649	0.92
Bubble sort 1	120349649	480744147	5.86
Bubble sort 2	120349649	241989000	3.49
Bubble sort 3	120349649	241978269	3.56
Shell sort 1	379091	653934	0.01
Shell sort 2	379091	653934	0.02
Dimenzija niza: 23000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	104823438	264488500	3.27

Selection sort 2	22993	264488500	1.90
Insertion sort 1	133340172	133363165	2.29
Insertion sort 2	88908780	133340172	1.00
Bubble sort 1	133340172	527758053	6.39
Bubble sort 2	133340172	264488500	3.78
Bubble sort 3	133340172	264487122	3.87
Shell sort 1	381524	668414	0.01
Shell sort 2	381524	668414	0.02

Dimenzija niza: 24000 elemenata

Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	111924449	287988000	3.49
Selection sort 2	23990	287988000	2.04
Insertion sort 1	144120035	144144022	2.49
Insertion sort 2	96096022	144120035	1.10
Bubble sort 1	144120035	575304028	6.91
Bubble sort 2	144120035	287988000	4.14
Bubble sort 3	144120035	287987622	4.20
Shell sort 1	452493	752351	0.01
Shell sort 2	452493	752351	0.02

Dimenzija niza: 25000 elemenata

Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	121059948	312487500	3.82
Selection sort 2	24988	312487500	2.20
Insertion sort 1	155932251	155957245	2.70
Insertion sort 2	103971500	155932251	1.18
Bubble sort 1	155932251	622650093	7.32
Bubble sort 2	155932251	312487500	4.30
Bubble sort 3	155932251	312483222	4.67
Shell sort 1	517347	829437	0.01

Shell sort 2	517347	829437	0.02
Dimenzija niza: 26000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	128269446	337987000	4.13
Selection sort 2	25987	337987000	2.37
Insertion sort 1	168464625	168490615	2.86
Insertion sort 2	112327082	168464625	1.27
Bubble sort 1	168464625	667784315	8.07
Bubble sort 2	168464625	337987000	4.81
Bubble sort 3	168464625	337937545	4.94
Shell sort 1	487064	811879	0.02
Shell sort 2	487064	811879	0.01
Dimenzija niza: 27000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	138212839	364486500	4.40
Selection sort 2	26992	364486500	2.59
Insertion sort 1	183682183	183709168	3.08
Insertion sort 2	122472788	183682183	1.41
Bubble sort 1	183682183	719793340	8.90
Bubble sort 2	183682183	364486500	5.25
Bubble sort 3	183682183	364428870	5.38
Shell sort 1	490298	827075	0.02
Shell sort 2	490298	827075	0.01
Dimenzija niza: 28000 elemenata			
Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	147165708	391986000	4.74
Selection sort 2	27987	391986000	2.74
Insertion sort 1	196345133	196373123	3.38

Insertion sort 2	130915421	196345133	1.52
Bubble sort 1	196345133	778876182	9.39
Bubble sort 2	196345133	391986000	5.58
Bubble sort 3	196345133	391969529	5.65
Shell sort 1	509237	859061	0.01
Shell sort 2	509237	859061	0.02

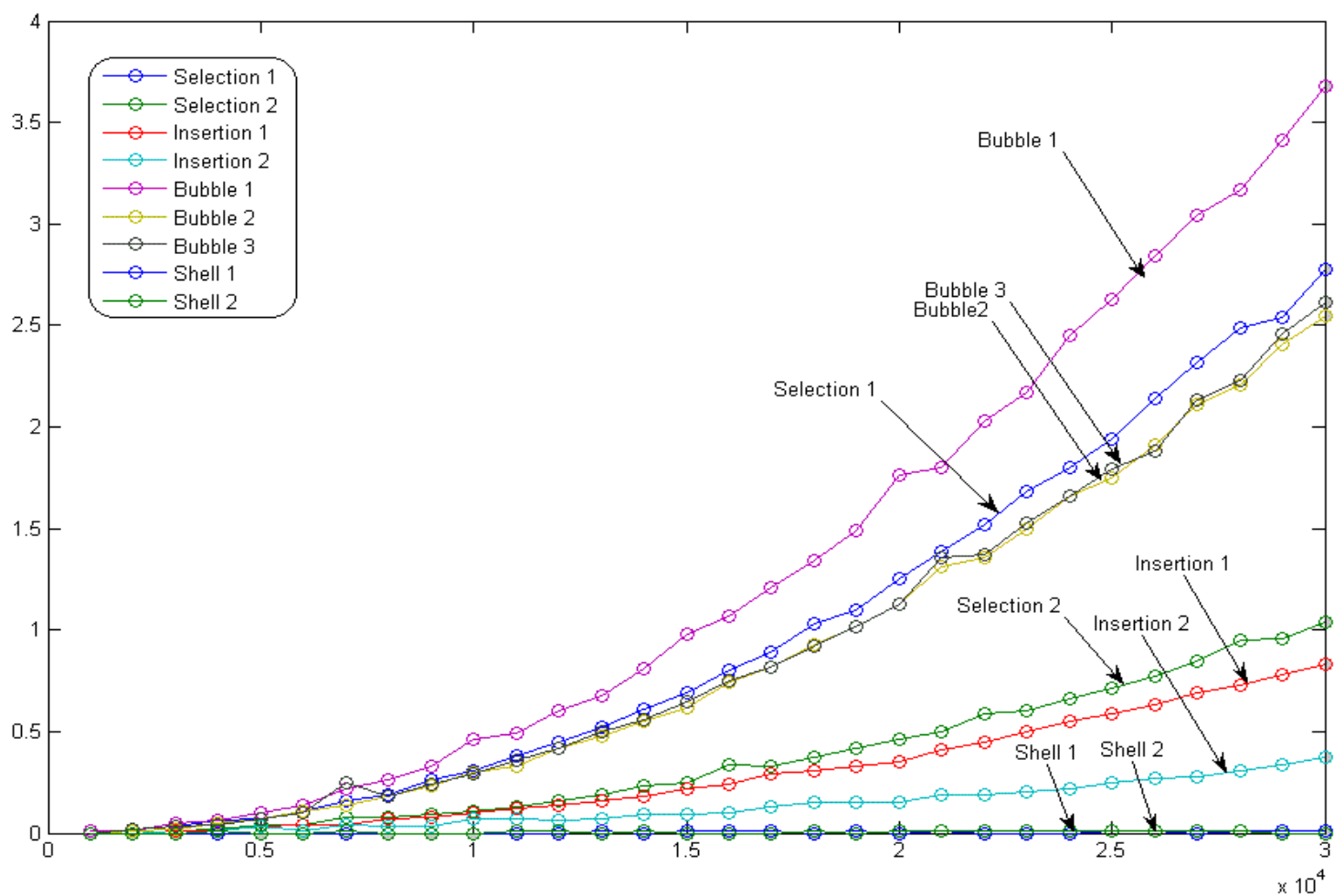
Dimenzija niza: 29000 elemenata

Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	155205807	420485500	5.15
Selection sort 2	28994	420485500	2.99
Insertion sort 1	208817403	208846393	3.63
Insertion sort 2	139230934	208817403	1.58
Bubble sort 1	208817403	834649218	10.14
Bubble sort 2	208817403	420485500	6.01
Bubble sort 3	208817403	420461847	6.06
Shell sort 1	576924	938891	0.01
Shell sort 2	576924	938891	0.02

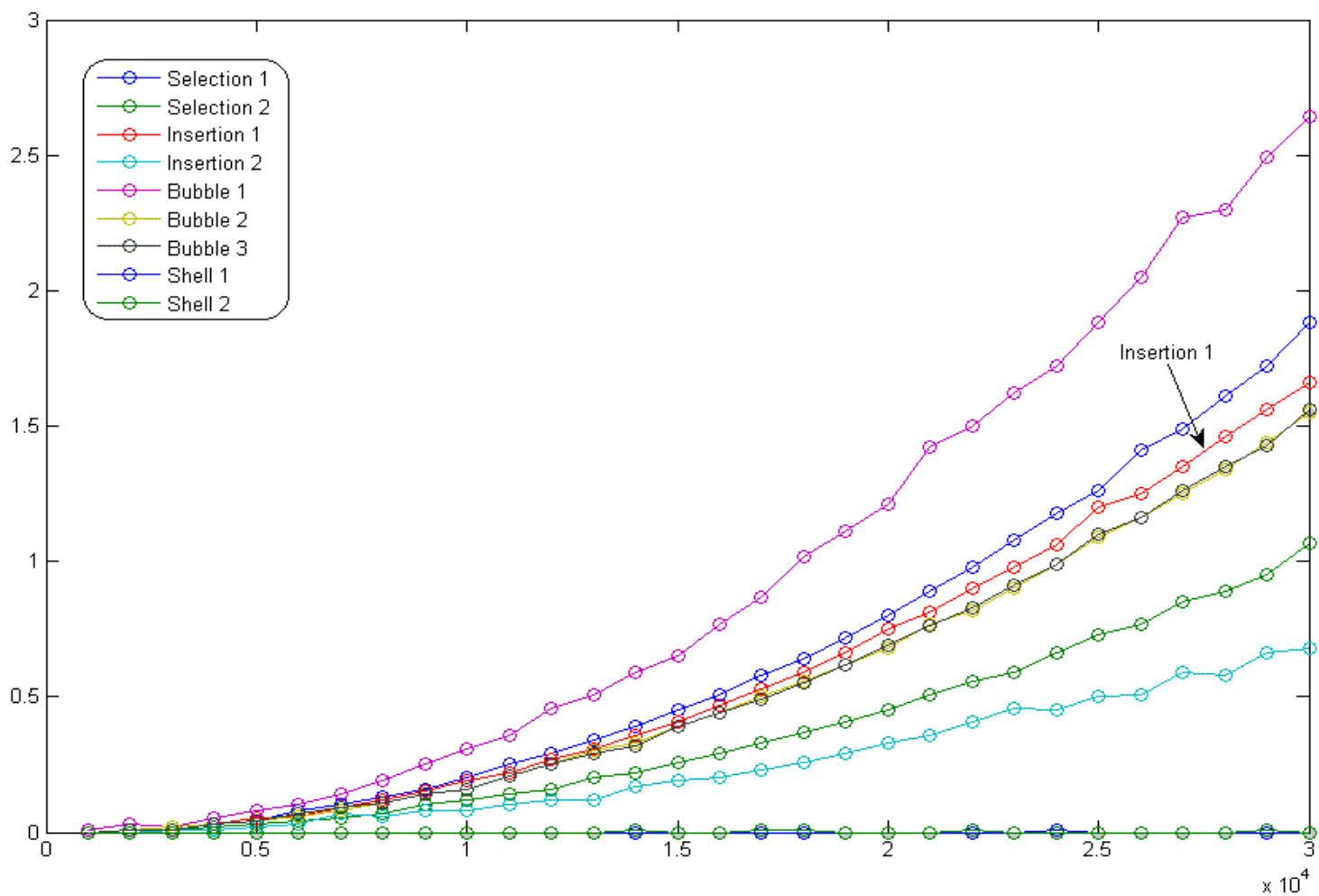
Dimenzija niza: 30000 elemenata

Algoritam	Zamena	Poređenja	Vreme
Selection sort 1	166216908	449985000	5.42
Selection sort 2	29986	449985000	3.19
Insertion sort 1	225499155	225529144	3.86
Insertion sort 2	150352769	225499155	1.70
Bubble sort 1	225499155	892830238	10.79
Bubble sort 2	225499155	449985000	6.47
Bubble sort 3	225499155	449956797	6.56
Shell sort 1	575655	950323	0.01
Shell sort 2	575655	950323	0.03

Grafici - eksperimentalni test



Sortiranje nizova slučajno generisanih brojeva



Sortiranje obrnuto sortiranih nizova

