

JEKON - Domaći rad - Upute

Ivan Slapničar

Split, 17. listopada 2006.

Sadržaj

1	Uvod	1
2	Opis algoritma <i>?? sort</i>	2
3	Usporedba s <i>Bubble sort</i> algoritmom	2
4	Zaključak	3

1 Uvod

Slijedi nekoliko adresa na kojima se obrađuju algoritmi za sortiranje. Uglavnom se spominju *Bubble sort* (koji smo pokazali u Dubrovniku), Heap sort, Insertion sort, Merge sort, Quick sort, Selection sort i Shell sort. Ima i primjera koda.

Zadatak je

- naći opis nekog od ovih algoritama na Internetu,
- programirati algoritam u Matlabu,
- usporediti algoritam s *Bubble sort* algoritmom koji je obrađen na predavanjima,
- napisati izvještaj u L^AT_EX-u - opisati novi algoritam, dati Vaš program, usporediti algoritme teoretski (složenost) i dati rezultate numeričkog testiranja.

Web adrese se nalaze u literaturi. Literatura se citira naredbom `\cite`: vidi, na primjer, [2].

O ovoj tematici se na Internetu može pronaći dosta materijala pa Vam neće biti problem pronaći dodatnu literaturu ako nešto nije dobro objašnjeno na ovim adresama.

Damir i ja Vam stojimo na raspolaganju.

2 Opis algoritma *?? sort*

Formule u tekstu se pišu ovako $x^2 + y^2 = 4$, a izdvojene formule ovako:

$$x^2 + y^2 = 4.$$

Matrice s pišu velikom slovima, $A, B, \dots$. Primijetimo i posebnu naredbu za tri točke.

Grčka slova se pišu vrlo logično: $\alpha, \beta, \gamma, \Gamma, \Xi, \dots$

Jednadžba može biti i numerirana i imati labelu te se kasnije može referencirati pomoću te labele:

$$x^3 - y^2 = \phi \xi \mu \eta \quad \rho \quad \delta. \quad (1)$$

Primijetimo se kako u jednadžbi (1) u poglavlju 2 postižu mali, veći, još veći i najveći razmak.

3 Usporedba s *Bubble sort* algoritmom

Algoritam *?? sort* opisan u poglavlju 2 usporedit ćemo s *Bubble sort* algoritmom koji glasi:

```
function A=bubbleSort (A)
    n = length(A);           % broj izvrsavanja:
    for i = n:-1:1           % n
        for j=1:i-1         % (n-1)+(n-2)+...+1=n(n-1)/2
            if A(j) > A(j+1)
                tmp = A(j);   % broj zamjena ovisi o tome
                A(j) = A(i);   % koliko je cesto zadovoljen
                A(i) = tmp;     % uvjet A(j)>A(j+1)
            end
        end
    end
end                           % Ukupan broj operacija ~n^2
```

Algoritme smo usporedili na sljedećim nizovima brojeva za $n = 200$ i $n = 400$:

1. niz slučajnih brojeva generiran naredbom `a=rand(n,1)`,
2. niz rastuće poredanih brojeva `a=1:n`,
3. niz padajuće poredanih brojeva `a=n:-1:1`,

Rezultati računanja (vrijeme) se nalaze u tablici¹.

¹Za detalje o formatiranju tablica pogledajte neka od L^AT_EX uputstava.

	<i>Bubble sort</i>		<i>?? sort</i>	
	$n = 200$	$n = 400$	$n = 200$	$n = 400$
<code>a=rand(n,1)</code>	1.12	1.12	1.12	1.12
<code>a=1:n</code>	1.12	1.12	1.12	1.12
<code>a=n:-1:1</code>	1.12	1.12	1.12	1.12

Tablica 1: Usporedba algoritama

4 Zaključak

Iz teoretske i praktične usporedbi algoritama provedenih u poglavljima 2 i 3 vidimo ...

Literatura

- [1] <http://linux.wku.edu/~lamonml/algor/sort/sort.html>
- [2] http://en.wikipedia.org/wiki/Category:Sort_algorithms
- [3] <http://www.aihorizon.com/essays/basiccs/lists/sorting/index.htm>
- [4] <http://www.softpanorama.org/Algorithms/sorting.shtml>