

Laborator 9 - Sortări simple-0

7.1. Sortarea prin inserție

Metoda sortării unui șir prin inserție reprezintă un algoritm simplu de sortare prin comparare. Șirul inițial, de lungime n , este ordonat în $n - 1$ pași, adăugând succesiv, câte un element la un subșir deja sortat. Pentru șiruri "aproape" ordonate sau de dimensiune redusă, acest algoritm este eficient din punct de vedere al timpului de execuție.

Problema:

Sortarea crescătoare a celor n elemente $v[0], v[1], \dots, v[n-1]$ ale unui vector v .

Algoritm:

1. Considerăm subșirul $v[0]$, care este deja ordonat.
2. Pentru fiecare i de la 1 la $n - 1$, subșirul $v[0]; v[1]; \dots; v[i - 1]$ este sortat și se caută poziția corespunzătoare a elementului $v[i]$: acel $k \in \{0; 1; \dots; i - 1\}$, cu proprietatea că $v[k - 1] \leq v[i] < v[k]$; elementele $v[k]; \dots; v[i - 1]$ se mută cu o poziție la dreapta, iar elementul curent, $v[i]$, se inserează pe poziția k .

Complexitatea algoritmului:

Op	Algorithm	Cost	Nr.rep.
1	for $i \leftarrow 2, n$ do		
2	$aux \leftarrow v[i]$		
3	$j \leftarrow i - 1$		
4	while $aux < v[j]$ <i>AND</i> $j \geq 0$ do		
5	$v[j + 1] \leftarrow v[j]$		
6	$j \leftarrow j - 1$		
	end while		
7	$v[j + 1] \leftarrow aux$		
	end for		

Calculati $T(n)$

Analizati (daca e cazul) complexitatea.

7.2 Sortarea prin selecția minimului

Este tot o sortare prin comparație.

Problema:

Sortarea crescătoare a celor n elemente $v[0], v[1], \dots, v[n-1]$ ale unui vector v , pentru fiecare i de la 0 la $n - 2$.

Algoritm:

1. Se caută valoarea minimă din șirul $v[i]; \dots; v[n-1]$, rămas neordonat.
2. Valoarea minimului se interschimbă cu elementul de la poziția i .

Complexitatea algoritmului:

Op.	Algoritm	Cost	Nr. rep.
1	Selectie($v[], n$)		
	for $i \leftarrow 1, n - 1$ do		
2	$poz_min \leftarrow i$		
3	for $j \leftarrow i + 1, n$ do		
4	if $v[j] < v[poz_min]$ then		
5	$poz_min \leftarrow j$		
	end if		
	end for		
6	if $poz_min \neq i$ then		
7	$v[poz_min] \leftrightarrow v[i]$		

[Get the mobile app](#)