ALGORIMI DE SORTARE

Profesor de informatică, *Csifó László Loránd*,
Colegiul Național "Unirea",
Tîrgu Mureș, Mureș

Vom analiza în articolul de faţă metodele cele mai uzuale de sortare. Încercăm să arătăm prin exemple cum se efectuează fiecare algoritm şi cât de eficienți sunt.

Metoda inserţiei directe INSERTSORT

Tabloul este văzut ca fiind format din două subtablouri a[1], a[2], ..., a[i-1] şi a[i], a[i+1], ..., a[N] (i=2, N). Secvenţa a[1], ..., a[i-1] este ordonată şi urmează ca a[i] să fie inserat în aceasta secvenţă la locul potrivit, astfel încât secvenţa a[1], ..., a[i-1], a[i] să rămână ordonată, urmând că în pasul următor cele două subtablouri considerate să fie a[1], ..., a[i] şi a[i+1], ..., a[N].

```
procedure InsertSort;
var
   i, j, aux: integer;
begin
   for i := 2 to n do begin
      aux: = a[i]; j := i - 1;
      while (aux < a[j]) and (j > 0) do begin
         a[j + 1] := a[j]; dec(j);
      end;
      a[j + 1] := aux;
   end;
end;
```

Semnificaţii tabel: **1)Gri 20% compar, 2)Gri 50% compar şi schimb, 3)Bold sortate**

i	j	j+1	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	aux	comp	swap
			8	4	3	1	15	9			
2	1	2	8	4	3	1	15	9	4	1	1
3	0	1	8	8	3	1	15	9	4	2	2
3	2	3	4	8	3	1	15	9	3	3	3
3	1	2	4	8	8	1	15	9	3	4	4
3	0	1	4	4	8	1	15	9	3	5	5
4	3	4	3	4	8	1	15	9	1	6	6
4	2	3	3	4	8	8	15	9	1	7	7
4	1	2	3	4	4	8	15	9	1	8	8
4	0	1	3	3	4	8	15	9	1	9	9
5	4	5	1	3	4	8	15	9	15	10	10
6	5	6	1	3	4	8	15	9	9	11	11
6	4	5	1	3	4	8	15	15	9	12	12
			1	**3**	**4**	**8**	**9**	**15**			

- Interschimbări făcute: 12
- Comparări făcute: 12
- Număr de operaţii: 12*2+12=36

Metoda selectării minimului MINSORT

Se consideră subtabloul a[i], ..., a[N], se caută elementul cu cheia minimă din acest subtablou şi apoi se interschimbă acest element cu elementul a[i], repetându-se procedeul pentru valori ale lui i de la 1 la N-1. Variabila minp reţine poziţia elementului minim în subtabloul a[i], ..., a[N].

```
procedure MinSort;
var
   i, j, aux, minp: integer;
begin
   for i := 1 to n - 1 do begin
      minp := i;
      for j := i + 1 to n do
         if a[j] < a[minp] then
            minp := j;
      aux := a[i]; a[i] := a[minp]; a[minp] := aux;
   end;
end;
```

Semnificaţii tabel: **1)Gri 20% compar, 2)Gri 50% compar şi schimb, 3)Bold sortate**

i	j	minp	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	aux	comp	swap
			8	4	3	1	15	9			
1	2	1	8	4	3	1	15	9	-	1	-
1	3	2	8	4	3	1	15	9	-	2	-
1	4	3	8	4	3	1	15	9	-	3	-
1	5	4	8	4	3	1	15	9	-	4	-
1	6	4	8	4	3	1	15	9	-	5	-
1	6	4	8	4	3	1	15	9	8	5	1
2	3	2	**1**	4	3	8	15	9	8	6	1
2	4	3	**1**	4	3	8	15	9	8	7	1
2	5	3	**1**	4	3	8	15	9	8	8	1
2	6	3	**1**	4	3	8	15	9	8	9	1
2	6	3	**1**	4	3	8	15	9	4	9	2
3	4	3	**1**	**3**	4	8	15	9	4	10	2
3	5	3	**1**	**3**	4	8	15	9	4	11	2
3	6	3	**1**	**3**	4	8	15	9	4	12	2
3	6	3	**1**	**3**	4	8	15	9	4	12	3
4	5	4	**1**	**3**	**4**	8	15	9	4	13	3
4	6	4	**1**	**3**	**4**	8	15	9	4	14	3
4	6	4	**1**	**3**	**4**	8	15	9	8	14	4
5	6	5	**1**	**3**	**4**	**8**	15	9	8	15	4
5	6	6	**1**	**3**	**4**	**8**	15	9	15	15	5
			1	**3**	**4**	**8**	**9**	**15**			

- Interschimbări făcute: 5
- Comparări făcute: 15
- Număr de operaţii: 5*3+15=30

Metoda selecţiei directe SELECTSORT

Se consideră subtabloul a[i+1], ..., a[N] care se parcurge de la stânga spre dreapta, comparând şi interschimbând perechile de elemente a[i] şi a[j] (j=i+1, N) care nu satisfac relaţia de ordine, procedeul repetându-se pentru i=1, N-1. Practic, la o parcurgere a subtabloului a[i+1], ..., a[N] are loc deplasarea elementului minim al acestui subtablou în poziţia a[i].

```
procedure SelectSort;
var
   i, j, aux:integer;
begin
   for i := 1 to n - 1 do
      for j := i + 1 to n do
         if a[i] > a[j] then begin
            aux := a[i]; a[i] := a[j]; a[j] := aux;
         end;
end;
```

Semnificaţii tabel: **1)Gri 20% compar, 2)Gri 50% compar şi schimb, 3)Bold sortate**

i	j	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	aux	comp	swap
		8	4	3	1	15	9			
1	2	8	4	3	1	15	9	8	1	1
1	3	4	8	3	1	15	9	4	2	2
1	4	3	8	4	1	15	9	3	3	3
1	5	1	8	4	3	15	9	3	4	3
1	6	1	8	4	3	15	9	3	5	3
2	3	**1**	8	4	3	15	9	8	6	4
2	4	**1**	4	8	3	15	9	4	7	5
2	5	**1**	3	8	4	15	9	4	8	5
2	6	**1**	3	8	4	15	9	4	9	5
3	4	**1**	**3**	8	4	15	9	8	10	6
3	5	**1**	**3**	4	8	15	9	8	11	6
3	6	**1**	**3**	4	8	15	9	8	12	6
4	5	**1**	**3**	**4**	8	15	9	8	13	6
4	6	**1**	**3**	**4**	8	15	9	8	14	6
5	6	**1**	**3**	**4**	**8**	15	9	15	15	7
		1	**3**	**4**	**8**	**9**	**15**			

- Interschimbări făcute: 7
- Comparări făcute: 15
- Număr de operaţii: 7*3+15=36

Metoda bulelor BUBBLESORT

Succesul algoritmului este asigurat de trecerea succesivă prin tablou, până când acesta este sortat, cu pecificaţia că, la fiecare trecere, elementele succesive i şi i+1 pentru care a[i]>a[i+1], vor fi interschimbate.

Metoda poate fi imbunătăţită dacă, după fiecare trecere, se va reţine ultima poziţie din tablou în care a avut loc o interschimbare, iar trecerea următoare se va efectua doar pînă la acea poziţie. În cazul în care la o trecere nu a avut loc nici o interschimbare algoritmul se va incheia.

```
procedure BubbleSort;
var
   i, j, aux: integer; ok: boolean;
begin
   repeat
      ok := true;
      for i := 1 to n - 1 do
         if a[i] > a[i + 1] then begin
            aux := a[i]; a[i] := a[i + 1]; a[i + 1] := aux; ok := false;
         end;
   until ok;
end;
```

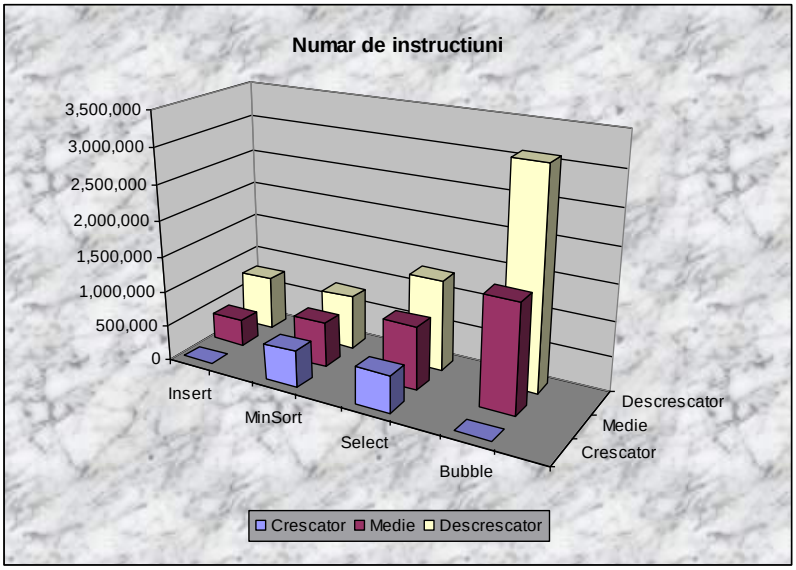
Semnificaţii tabel: **1)Gri 20% comparare, 2)Gri 50% interschimbare, 3)Bold sortate**

i	i+1	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	aux	comp	swap
		8	4	3	1	15	9			
1	2	8	4	3	1	15	9	8	1	1
2	3	4	8	3	1	15	9	8	2	2
3	4	4	3	8	1	15	9	8	3	3
4	5	4	3	1	8	15	9	8	4	3
5	6	4	3	1	8	15	9	15	5	4
1	2	4	3	1	8	9	**15**	4	6	5
2	3	3	4	1	8	9	15	4	7	5
3	4	3	1	4	8	9	15	4	8	5
4	5	3	1	4	8	9	**15**	4	9	5
5	6	3	1	4	8	9	**15**	4	10	5
1	2	3	1	4	8	**9**	**15**	3	11	6
2	3	1	3	4	8	**9**	**15**	3	12	6
3	4	1	3	4	8	**9**	**15**	3	13	6
4	5	1	3	4	8	**9**	**15**	3	14	6
5	6	1	3	4	8	**9**	**15**	3	15	6
1	2	1	3	4	**8**	**9**	**15**	3	16	6
2	3	1	3	4	**8**	**9**	**15**	3	17	6
3	4	1	3	4	**8**	**9**	**15**	3	18	6
4	5	1	3	4	**8**	**9**	**15**	3	19	6
5	6	1	3	4	**8**	**9**	**15**	3	20	6
		1	**3**	**4**	**8**	**9**	**15**			

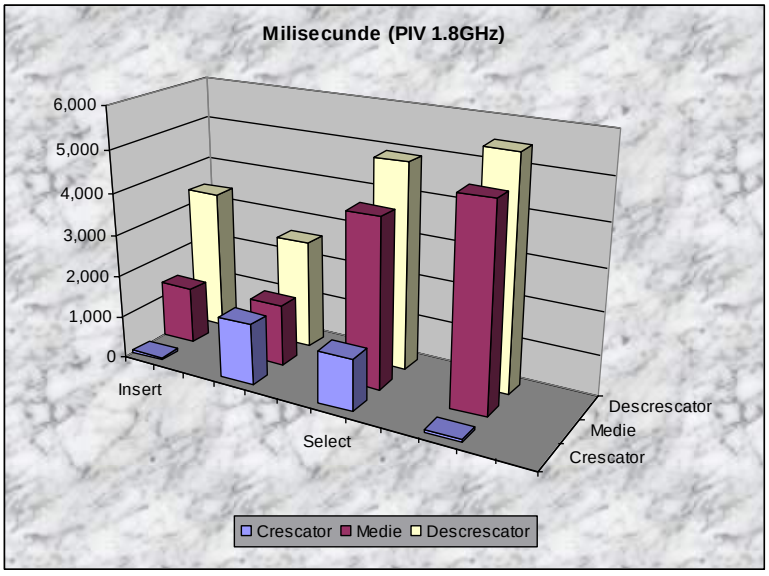
- Interschimbări făcute: 6
- Comparări făcute: 20
- Număr de operaţii: 6*4+20=44

Metode cu complexitate O(N^2) COMPARARE

Nr	Insert		MinSort		Select		Bubble	
	Crescator	Descresc	Crescator	Descresc	Crescator	Descresc	Crescator	Descresc
	5n-4	$(3n^2+7n-8)/2$	n^2+3n-3	$(3n^2+9n-4)/2$	n^2	$(5n^2-3n-2)/2$	2(n-1)	$6n^2-4n-2$
10,000	50	150,035	100,030	150,045	100,000	249,985	20	599,960
17,500	87	459,436	306,302	459,454	306,250	765,599	35	1,837,430
25,000	125	937,587	625,075	937,612	625,000	1,562,462	50	3,749,900
32,000	160	1,536,112	1,024,096	1,536,144	1,024,000	2,559,952	64	6,143,872
Cresc	**106**		**513,876**		**513,813**		**42**	
Medie	**385,449**		**642,345**		**899,156**		**1,541,416**	
Descre	**770,793**		**770,814**		**1,284,500**		**3,082,790**	



Nr	Insert			MinSort			Select			Bubble		
	Cresc	Aleator	Descresc	Cresc	Aleator	Descresc	Cresc	Aleator	Descresc	Cresc	Aleator	Descresc
10,000	55	275	550	330	275	550	275	770	935	55	990	1,045
17,500	55	825	1,595	825	880	1,540	770	2,420	2,970	55	2,970	3,245
25,000	55	1,650	3,245	1,760	1,760	3,135	1,485	4,895	5,995	55	5,775	7,095
32,000	55	2,640	5,280	2,915	2,860	5,170	2,420	8,030	9,735	55	9,570	10,615
Cresc	**55**			**1,458**			**1,238**			**55**		
Medie	**1,348**			**1,444**			**4,029**			**4,826**		
Descr	**3,373**			**2,599**			**4,909**			**5,500**		



Bibliografie

1. Thomas H. Cormen, Charles H. Leiserson, Ronald R. Rivest: Introducere în algoritmi, editura Computer Libris Agora, București, 2006
2. Simona Haidu, Dana Vaida, Eugen Ionescu: Informatică (pentru grupele de performanţă) clasa a IX-a, editura Dacia Educațional, Cluj-Napoca, 2004
3. Bogdan Pătruţ: Învăţaţi limbajul Pascal în 12 lecţii, editura Teora, București, 1999
4. Tudor Sorin: Învăţaţi limbajul C++ Standard. Curs pentru clasele a IX-a și a X-a, , editura L&S INFO-MAT, București, 2008
5. Mariana Miloșescu: Informatică, profilul real. Manual pentru clasa a IX-a, editura Didactică și Pedagogică, București, 2006