

Software pentru ordonarea multirang a componentelor unei colectivități

Se prezintă un algoritm pentru dispunerea elementelor unei colectivități în raport cu mai multe criterii și un software aferent, programul moscs.cpp, care se află la adresa www.moscs.ase.ro

1. Ordonarea după o caracteristică

Se consideră o mulțime A formată din elementele a_1, \dots, a_n . Pentru descrierea elementelor sunt utilizate caracteristicile c_1, \dots, c_m . Fiecare element $a_i, i=1,2,3, \dots, n$ se descrie cu nivelul α_{ij} al caracteristicii c_i .

Dintre caracteristicile c_1, \dots, c_m este selectată

caracteristica c_r , numită criteriu de ordonare, astfel încât elementele mulțimii A se dispun unele după altele, conform cu valoarea caracteristicii alese.

Pentru exemplificare, se consideră mulțimea întreprinderilor ale căror caracteristici sunt date în tabelul 1, ce va fi afișat după rularea programului. Funcția care face afișarea întreprinderilor și caracteristicilor se numește *af_intre()*. Prototipul funcției este următorul: *void af_intre(intrepr t)*, unde t este o variabilă de tip **intrepr**.

Rangul este un număr asociat elementului a_i , prin care se indică poziția acestuia într-o listă de elemente, dispuse în raport cu un criteriu.

Datele inițiale ale problemei

Tabelul 1

Codul	Denumire	Cifra de afaceri	Număr muncitori	Productivitate
AA	METEOR	300	100	20
AB	FAURAR	100	300	200
AC	RATING	90	80	20
AD	BOOGYE	200	30	20
AE	RATION	100	800	400
AF	SC TARO	1000	20	400
AG	FLAMENCO	20	20	30
AH	PC UNIT	10	10	2
AI	COMPUTER	10	400	20
AJ	FABY	100	100	100

Dacă se consideră drept criteriu **codul întreprinderii**, definit astfel: **char cod[5]**, dispunerea întreprinderilor se face crescător, în funcție de valoarea codului fiecărei întreprinderi, fapt ce duce la obținerea rangurilor din tabelul 2.

Funcția care face ordonarea întreprinderilor după câmpul **cod** este *ord_cod()*. Prototipul funcției este următorul: *intrepr ord_cod(intrepr intr)*, cu **intr** o variabilă de tipul **intrepr**.

Dispunerea întreprinderilor după cod

Tabelul 2

Rang	Cod	Denumire
1	AA	METEOR
2	AB	FAURAR
3	AC	RATION
4	AD	BOOGYE
5	AE	RATION
6	AF	SC TARO
7	AG	FLAMENCO
8	AH	PC UNIT
9	AI	COMPUTER
10	AJ	FABY

Management

Când se consideră criteriu de ordonare **denumirea întreprinderii**, definită **char den[40]**, dispunerea întreprinderilor se face crescător, în funcție de valoarea caracteristicii **denumire** a fiecărei întreprinderi, și duce la obținerea rangurilor din tabelul 3. Funcția care face

ordonarea întreprinderilor după caracteristica **denumire** este *ord_den()*.

Prototipul funcției este: *void ord_den(intrepr intr1)*, cu **intr1** o variabilă de tipul **intr**.

Dispunerea întreprinderilor după denumire

Tabelul 3

Rang	Cod	Denumire
1	AD	BOOGYE
2	AI	COMPUTER
3	AJ	FABY
4	AB	FAURAR
5	AG	FLAMENCO
6	AA	METEOR
7	AH	PC UNIT
8	AC	RATING
9	AE	RATION
10	AF	SC TARO

Dacă se consideră întreprinderile după **cifra de afaceri**, definită **float cifa**, dispunerea se face în ordine descrescătoare, obținându-se rangurile din tabelul 4.

Funcția care face ordonarea întreprinderilor după caracteristica **cifra de afaceri** este *ord_cifa()*. Prototipul funcției este: *void ord_cifa(intrepr intr1)*.

Dispunerea întreprinderilor după cifra de afaceri

Tabelul 4

Cod	Denumire	Rang	Cifra de afaceri	Rang_calc
AF	SC TARO	1	1000	1
AA	METEOR	2	300	70
AD	BOOGYE	3	200	80
AB	FAURAR	4	100	90
AE	RATION	5	100	91
AJ	FABY	6	100	92
AC	RATING	7	90	93
AG	FLAMENCO	8	20	98
AH	PC UNIT	9	10	99
AI	COMPUTER	10	10	100

În cazul în care se consideră întreprinderile după caracteristica **numărul de muncitori**, definită **int nr_m**, dispunerea se face în ordine descrescătoare, obținându-se rangurile din tabelul 5. Funcția care face

ordonarea întreprinderilor după caracteristica **numărul muncitorilor** este *ord_nrm()*. Prototipul funcției este: *void ord_nrm(intrepr intr1)*.

Dispunerea întreprinderilor după numărul de muncitori

Tabelul 5

Rang	Rang_calculat	Cod	Denumire	Nr. muncitori
1	1	AE	RATION	800
2	40	AI	COMPUTER	400
3	50	AB	FAURAR	300
4	70	AA	METEOR	100
5	71	AJ	FABY	100
6	72	AC	RATING	80
7	77	AD	BOOGYE	30
8	78	AF	SC TARO	20
9	79	AG	FLAMENCO	20
10	80	AH	PC UNIT	10

Management

Când se consideră întreprinderile după câmpul **productivitate**, definit **float prod**, dispunerea se efectuează în ordine descrescătoare, obținându-se rangurile din tabelul

6. Funcția care face ordonarea întreprinderilor după caracteristica productivitate este *ord_prod()*. Prototipul funcției este: *void ord_prod(intrepr intr1)*.

Dispunerea întreprinderilor după productivitate

Tabelul 6

Rang	Rang_calculat	Cod	Denumire	Productivitate
1	1	AE	RATION	400
2	2	AF	SC TARO	400
3	20	AB	FAURAR	200
4	30	AJ	FABY	100
5	37	AG	FLAMENCO	30
6	38	AA	METEOR	20
7	39	AC	RATING	20
8	40	AD	BOOGYE	20
9	41	AI	COMPUTER	20
10	42	AH	PC UNIT	2

Variabila **rangul_calculat**, pentru caracteristicile **productivitate**, **număr muncitori**, **cifra de faceri**, rezultă din secvența:

```
if (x1[0] <= 100)
{
    poz [i] = x1[0]-x1[i]; if (poz [i]==0) poz [i]=1;
    for (int k=0; k<i; k++) if (poz [k]==poz [i]) poz [i]+=1;
}
else
{
    poz [i]=(x1[0]-x1[i])/10; if (poz [i]==0) poz [i]=1;
    for (int k=0; k<i; k++) if (poz [k]==poz [i]) poz [i]+=1;
}
```

unde:

x1[i] reprezintă valorile câmpurilor.

-----PENTRU A PĂRĂSI APLICAȚIA ALEGEȚI OPTIUNEA 0-----

OPȚIUNILE MENIULUI

- 1 : PRELUARE DATE DESPRE COLECTIVITATE
 - 2 : PRELUARE DATE DIN FIȘIER
 - 3 : NOI MEMBRI AI COLECTIVITĂȚII
 - 4 : NOI CARACTERISTICI ALE COLECTIVITĂȚII
 - 5 : AFIȘARE MEMBRI
 - 6 : AFIȘARE CARACTERISTICI
 - 7 : AFIȘARE SITUAȚII DUPĂ CARACTERISTICI
 - 8 : AFIȘARE SITUAȚIE FINALĂ
 - 9 : RANG AGREGAT PRIN SUMĂ
 - 10 : RANG AGREGAT PRIN PRODUS
 - 11 : RANG MEDIU CALCULAT
 - 12 : SALVARE DATE ÎN FIȘIER
- OPȚIUNEA DUMNEAVOASTRĂ

Figura 1 Meniul aplicației

Management

2. Ordonarea multirang

Atunci când se consideră două caracteristici și se dorește rangul agregat, există mai multe metode de a rezolva această problemă.

Pentru mulțimea întreprinderilor și pentru caracteristicile considerate, prin punerea rangurilor din tabelele 2, 3, 4, 5, 6 rezultă tabelul 7. Funcția ce are ca finalitate tabelul 7 este *af_final()*. Prototipul funcției este: *void af_final(intrepr h,int a[][10])*, unde **h** este o variabilă de tipul **intrepr**, iar **a** este o matrice de tip **int**.

Situația finală a rangurilor

Tabelul 7

Cod	Rangul după denumire	Rangul după cifra de afaceri	Rangul după nr. de muncitori	Rangul după productivitate
AA	5	2	4	7
AB	8	6	3	3
AC	7	7	6	8
AD	2	3	7	9
AE	9	4	1	1
AF	10	1	8	2
AG	4	8	9	5
AH	6	10	10	10
AI	3	9	2	6
AJ	1	5	5	4

Agregarea rangurilor prin însumare constă în adunarea rangurilor obținute după diferite clasificări și conduce la apartenența elementelor colectivității la clase de componente.

Prin ordonarea crescătoare a rangurilor agregate prin însumare, se obține tabelul 8. Funcția care face ordonarea

întreprinderilor după rangurile însumate este *ord_af_ins()*. Prototipul funcției este: *void ord_af_ins(intrepr intr,intrepr h,int aa[][10])*.

Programul calculează, de asemenea, rangul real pe care îl are întreprinderea respectivă, făcând abstracție de numărul întreprinderilor.

Disponerea întreprinderilor după rangul însumat

Tabelul 8

Cod	Denumirea	Rangul însumat	Rangul	Rangul calculat
AB	FAURAR	17	1	1
AA	METEOR	20	2	4
AE	RATION	22	3	6
AD	BOOGYE	24	4	8
AJ	FABY	26	5	10
AF	SC TARO	27	6	11
AI	COMPUTER	28	7	12
AC	RATING	32	8	16
AG	FLAMENCO	34	9	18
AH	PC UNIT	45	10	29

Rangul_calculat a fost obținut cu formula: $(x1[i]-x1[0])/min+1$, unde:

- ⇒ **min** reprezintă minimul calculat cu formula $if ((x1[i]-x1[i-1])<min)$
 $min = x1[i]-x1[i-1]$, pentru $i = 0, nr_intr-1$;
- ⇒ **x1[i]**, **x1[0]** sunt respectiv valoarea curentă și valoarea cea mai mică.

Agregarea produs este realizată prin înmulțirea valorilor rangurilor din tabelul 7, iar ordonarea crescătoare a rangului produs duce la obținerea tabelului 9. Funcția ce realizează ordonarea întreprinderilor în funcție de rangul produs este *ord_af_produs()*. Prototipul funcției este: *void ord_af_produs(intrepr intr,intrepr h,int a1[][10])*.

Management

Disponerea întreprinderilor după rangul produs

Tabelul 9

Codul	Denumirea	Rangul produs
AE	RATION	270
AA	METEOR	336
AB	FAURAR	360
AD	SC TARO	756
AF	FABY	960
AI	BOOGYE	1944
AJ	COMPUTER	2400
AC	RATING	8064
AG	FLAMENCO	12600
AH	PC UNIT	56000

3. Creșterea generalității aplicației

Utilizatorul are posibilitatea să execute rezolvarea oricărei probleme ce presupune ordonarea și calculul rangurilor.

În acest scop, se definesc următoarele date de intrare ale problemei:

- numele colectivității;
- caracteristicile colectivității (nu mai mult de nouă);
- tipul caracteristicilor (int sau float).

Pentru următorul stadiu al aplicației ne propunem introducerea variabilelor de tip **char**.

Unele dintre caracteristici au tendința de maximizare, pentru a genera efecte pozitive la nivelul elementelor colectivității. În acest caz, rangurile uzează ordonarea descrescătoare.

Dacă avem caracteristici pentru care se urmărește scăderea valorilor, vom ordona descrescător valorile. Pentru început, aplicația vizează doar obiective de maximum. În scopul de a realiza și obiective de minim, trebuie introduse noi proceduri care se atașează produsului existent.

Produsul este realizat în limbajul C, are 6 module, ocupă pe disc 32 KB, iar configurația minimă necesară va fi furnizată de un calculator echipat cu microprocesor 486, cu spațiu de RAM de 32 MB și un hard de 10 MB.

Concluzii

Implementarea acestui produs software, aflat la adresa www.moscs.ase.ro, permite noi abordări când se efectuează analize ale unor colectivități omogene de elemente și se

dorește extragerea unor submulțimi, prin agregarea de criterii. Direcțiile următoare ale cercetărilor trebuie să includă o combinație a algoritmilor de ordonare multirang cu algoritmi de ordonare în care sunt utilizați coeficienți de importanță. În mod corespunzător, pentru noua tipologie de algoritmi se dezvoltă un produs software suficient de flexibil.

Bibliografie

[BODEA02] Constanța-Nicoleta BODEA – *Handbook of Project-Based Management*, București, Editura Economică, 2002

[IVAN90] Ion IVAN - *Modele de ierarhizare a produselor*, Revista de Statistică, Nr. 6, 1990

[POPESCU00] Theodor POPESCU – *Serii de timp*, București, Editura Tehnică, 2000

[SMEUREANU01] Ion SMEUREANU – *Programarea în limbajul C/C++*, București, Editura Cison, 2001

[SMEUREANU02] Ion SMEUREANU – *Programarea orientată obiect în limbajul C++*, București, Editura Cison, 2002

[TROIE01] Liviu TROIE – *Analiza statistică a activității economice și a gestiunii financiare a întreprinderii*, București, Editura ASE, 2001

Prof. univ. dr. inf. ec. Ion IVAN
Academia de Studii Economice București,
Stud. Daniel MILODIN
CSIE, an IV