

Rendezések

Feladat

Rendezési algoritmusok kipróbálása, hatékonysági viselkedésének vizsgálata.

A rendezések egy ElemSzam méretű r tömbben történik.

Többféle föltöltés közül lehet választani:

- a növekvően / csökkenően rendezettől,
- a majdnem rendezetten át,
- véletlenszerűen kitöltöttig.

A föltöltés nemcsak az r-re vonatkozik, hanem az s-re is. Ez használható föl a további rendezések kiinduló értékének beállítására.

Keretprogram

```
1 Program RendezesKeret;  
2 {  
3   Rendezési algoritmusok kipróbálásához keret.  
4   A rendezések egy ElemSzam méretű r tömbben történjen.  
5   Többféle föltöltés közül lehet választani:  
6   - a növekvően / csökkenően rendezettől,  
7   - a majdnem rendezetten át,  
8   - véletlenszerűen kitöltöttig.  
9   A föltöltés nemcsak az r-re vonatkozik, hanem az s-re is.  
10  Ez használható föl a további rendezések kiinduló értékének  
11  beállítására.  
12 }  
13 Uses  
14   Newdelay, Crt, Dos;  
15 Const  
16   MaxN=1000;  
17 Type  
18   TElem=Integer;  
19   TSorozat=Array[1..MaxN] of TElem;  
20 Var  
21   Kezdet, Veg,  
22   ElteltIdo: LongInt;  
23   s, r      : TSorozat;   {A föltöltések mindkettőt beállítják,  
24                       a rendezések az r-t rendezik.}  
25   ElemSzam : Integer;  
26  
27   {SI RENDMENU.PAS}  
28   {  
29     Ebben a file-ban található a rendezéshez szükséges menüő program.  
30     Számunkra érdekes most ebből mindössze:  
31     - Menuzes(FoMenu), amely beindítja a párbeszédet és hívogatja  
32       a megfelelő rendező eljárásokat.  
33     - TombKiir, amely az r tartalmát oszloposan megjeleníti a képernyőn.  
34   }
```

```

35
36 Procedure OraIndul;
37   Var
38     o,p,mp,szmp: Word;
39     ol: LongInt;
40 Begin
41   GetTime(o,p,mp,szmp); ol:=o; {konverzió LongInt-té}
42   Kezdet:=szmp+100*(mp+60*(p+60*ol{konverzió LongInt-té}))
43 End; {OraIndul}
44
45 Procedure OraAll;
46   Var
47     o,p,mp,szmp: Word;
48     ol: LongInt;
49 Begin
50   GetTime(o,p,mp,szmp); ol:=o; {konverzió LongInt-té}
51   Veg:=szmp+100*(mp+60*(p+60*ol{konverzió LongInt-té}));
52   ElteltIdo:=Veg-Kezdet
53 End; {OraAll}
54
55 Procedure Kiirasok(cim: String);
56   Var
57     i: Integer;
58 Begin
59   ClrScr;
60   TombKiir(cim);
61   Writeln; Writeln('Eltelt idő:',ElteltIdo, ' századmásodperc.')
62 End; {Kiirasok}
63
64 Procedure EgyszeruCseres;
65 Begin
66   OraIndul;
67   {itt van az r rendezése}
68   OraAll;
69   Kiirasok('Egyszerű cserés');
70   r:=s; {Az eredeti értéket visszaállítja}
71 End; {EgyszeruCseres}
72
73 Procedure MinimumKivalasztasos;
74 Begin
75   OraIndul;
76   {itt van az r rendezése}
77   OraAll;
78   Kiirasok('Minimumkiválasztásos');
79   r:=s; {Az eredeti értéket visszaállítja}
80 End; {MinimumKivalasztasos}
81
82 Procedure Buborek;
83 Begin
84   OraIndul;
85   {itt van az r rendezése}
86   OraAll;
87   Kiirasok('Buborék');
88   r:=s; {Az eredeti értéket visszaállítja}
89 End; {Buborek}
90

```

```

91  Procedure JavitottBuborek;
92  Begin
93      OraIndul;
94      {itt van az r rendezése}
95      OraAll;
96      Kiirasok('Javitott buborék');
97      r:=s; {Az eredeti értéket visszaállítja}
98  End; {JavitottBuborek}
99
100 Procedure Beilleszteses;
101 Begin
102     OraIndul;
103     {itt van az r rendezése}
104     OraAll;
105     Kiirasok('Beillesztéses');
106     r:=s; {Az eredeti értéket visszaállítja}
107 End; {Beilleszteses}
108
109 Begin
110     ClrScr;
111     Randomize;      {Mindig véletlenszerű kezdőérték}
112     ElemSzam:=100; {Opcionális méret}
113     Menuzes (FoMenu);
114 End.

```

RendMenu.pas – include-állomány

```
{$f+ a saját eljárások, függvények kezeléséhez igényli a TURBO PASCAL}
Const
    MaxDb = 6;
Type
    MenuPont = Record
        kerdes: String;
        sorsz : Byte;
        tevek : Procedure;
        segit : Word;
    End;
    Menu = Record
        cim: String;
        db : Byte;
        am : Array [1..MaxDb] of MenuPont;
        bfx,bfy: Byte;
    End;

Procedure Varakozas; Forward;
Procedure Ures; Forward;
Procedure AblakRajzolas(bfx,bfy,hossz,szel: Byte); Forward;
Procedure Menuzes(m: Menu); Forward;
Procedure Vege; Forward;

{
    Menüdeklaráció:
}
Procedure Menu1; Forward; {Parmétermenü}
Procedure Menu2; Forward; {Rendezések menüje}
Procedure Menu3; Forward; {Föltöltések menüje}
Procedure Menu1_1; Forward; {Az egyes rendezéseket hívó eljárás}
Procedure Menu1_2; Forward; {Az egyes rendezéseket hívó eljárás}
Procedure Menu1_3; Forward; {Az egyes rendezéseket hívó eljárás}
Procedure Menu1_4; Forward; {Az egyes rendezéseket hívó eljárás}
Procedure Menu1_5; Forward; {Az egyes rendezéseket hívó eljárás}
Procedure FeltoltNovekedve; Forward;
Procedure FeltoltCsokkenve; Forward;
Procedure FeltoltAzonossal; Forward;
Procedure FeltoltVeletlennel; Forward;
Procedure FeltoltMajdnemRendezve; Forward;
{
    Segéd globális eljárás:
}
Procedure TombKiir(cim: String); Forward;
{
    A megirandó rendezések:
}
Procedure EgyszeruCseres; Forward;
Procedure MinimumKivalasztasos; Forward;
Procedure Buborek; Forward;
Procedure JavitottBuborek; Forward;
Procedure Beilleszteses; Forward;
```

```

{
    Menüleírótábla deklarációja:
}
Const
    FoMenu : Menu=(cim: 'Főmenü';
        db : 4;
        am : ((kerdes:'Paramétermegadás';sorsz:1;tevek:Menu1;segit:1),
            (kerdes:'Feltöltés';sorsz:1;tevek:Menu3;segit:3),
            (kerdes:'Rendezési módszer';sorsz:1;tevek:Menu2;segit:2),
            (kerdes:'Vége'; sorsz:1;tevek:Vege;segit:99),
            (kerdes:'';sorsz:0;tevek:Ures;segit:0),
            (kerdes:'';sorsz:0;tevek:Ures;segit:0));
        bfx: 10;
        bfy: 5);
    AllMenu: Menu=(cim: 'Rendezések';
        db : 6;
        am : ((kerdes:'Egyszerű cserés';sorsz:1;tevek:Menu1_1;segit:4),
            (kerdes:'Minimumkiválasztásos';sorsz:1;tevek:Menu1_2;segit:5),
            (kerdes:'Buborék';sorsz:1;tevek:Menu1_3;segit:6),
            (kerdes:'Javított buborék';sorsz:1;tevek:Menu1_4;segit:7),
            (kerdes:'Beillesztéses';sorsz:3;tevek:Menu1_5;segit:8),
            (kerdes:'Vége'; sorsz:1;tevek:Vege;segit:99));
        bfx: 10;
        bfy: 5);
    Al2Menu: Menu=(cim: 'Föltöltések';
        db : 6;
        am : ((kerdes:'Növekedően';sorsz:1;tevek:FeltoltNovekedve;segit:7),
            (kerdes:'Csökkenően';sorsz:1;tevek:FeltoltCsokkenve;segit:8),
            (kerdes:'Azonosakkal';sorsz:1;tevek:FeltoltAzonossal;segit:9),
            (kerdes:'Véletlennel';sorsz:3;tevek:FeltoltVeletlennel;segit:10),
            (kerdes:'Majdnem rendezve';
                sorsz:1;tevek:FeltoltMajdnemRendezve;segit:11),
            (kerdes:'Vége'; sorsz:1;tevek:Vege;segit:99));
        bfx: 10;
        bfy: 5);
{
    Menütevékenységek deklarációja:
}
Procedure Menu1;
    Var
        HolX: Integer;
    Begin
        AblakRajzolas(1,1,24,79);
        Repeat
            GotoXY(20,12); Write('Tömbméret (100..',MaxN,'):'); HolX:=WhereX;
            Write(' ');
            GotoXY(HolX,12); Readln(ElemSzam)
        Until (ElemSzam>=100) and (ElemSzam<=MaxN);
        ClrScr;
End; {1.Menü}

Procedure Menu2;
Begin
    AblakRajzolas(1,1,24,79);
    Menuzes(AllMenu);
End; {2.Menü}

Procedure Menu3;
Begin
    AblakRajzolas(1,1,24,79);
    GotoXY(38,12);
    Menuzes(Al2Menu);
End; {3.Menü}

```

```

Procedure Menu1_1;
Begin
  AblakRajzolas(1,1,24,79);
  GotoXY(28,12);
  Writeln('Egyszerű cserés rendezés');
  Window(2,2,78,23); {Az aktív ablakterület}
  EgyszeruCseres;
  Varakozas
End; {1.1.Menü}

```

```

Procedure Menu1_2;
Begin
  AblakRajzolas(1,1,24,79);
  GotoXY(26,12);
  Writeln('Minimumkiválasztásos rendezés');
  Window(2,2,78,23); {Az aktív ablakterület}
  MinimumKivalasztasos;
  Varakozas
End; {1.2.Menü}

```

```

Procedure Menu1_3;
Begin
  AblakRajzolas(1,1,24,79);
  GotoXY(36,12);
  Writeln('Buborék');
  Window(2,2,78,23); {Az aktív ablakterület}
  Buborek;
  Varakozas
End; {1.3.Menü}

```

```

Procedure Menu1_4;
Begin
  AblakRajzolas(1,1,24,79);
  GotoXY(32,12);
  Writeln('Javított buborék');
  Window(2,2,78,23); {Az aktív ablakterület}
  JavitottBuborek;
  Varakozas
End; {1.4.Menü}

```

```

Procedure Menu1_5;
Begin
  AblakRajzolas(1,1,24,79);
  GotoXY(36,12);
  Writeln('Beillesztéses');
  Window(2,2,78,23); {Az aktív ablakterület}
  Beilleszteses;
  Varakozas
End; {1.5.Menü}

```

```

Procedure FeltoltNovekedve;
  Var
    i: Integer;
Begin
  AblakRajzolas(1,1,24,79);
  s[1]:=Random(ElemSzam);
  For i:=2 to ElemSzam do
    Begin
      s[i]:=s[i-1]+Random(3)
    End;
  r:=s;
  TombKiir('Föltöltés növekedve');

```

```

    Varakozas
End; {FeltoltNovekedve}

Procedure FeltoltCsokkenve;
    Var
        i: Integer;
    Begin
        AblakRajzolas(1,1,24,79);
        s[1]:=Random(ElemSzam);
        For i:=2 to ElemSzam do
            Begin
                s[i]:=s[i-1]-Random(3)
            End;
            r:=s;
            TombKiir('Föltöltés csökkenve');
            Varakozas
        End; {FeltoltCsokkenve}

Procedure FeltoltAzonossal;
    Var
        i: Integer;
    Begin
        AblakRajzolas(1,1,24,79);
        s[1]:=Random(ElemSzam);
        For i:=2 to ElemSzam do
            Begin
                s[i]:=s[i-1]
            End;
            r:=s;
            TombKiir('Föltöltés azonosakkal');
            Varakozas
        End; {FeltoltAzonossal}

Procedure FeltoltVeletlennel;
    Var
        i: Integer;
    Begin
        AblakRajzolas(1,1,24,79);
        For i:=1 to ElemSzam do
            Begin
                s[i]:=Random(ElemSzam)
            End;
            r:=s;
            TombKiir('Föltöltés véletlenekkel');
            Varakozas
        End; {FeltoltVeletlennel}

Procedure FeltoltMajdnemRendezve;
    Var
        i,j,k: Integer;
        seged: TElem;
    Begin
        AblakRajzolas(1,1,24,79);
        s[1]:=Random(ElemSzam);
        For i:=2 to ElemSzam do
            Begin
                s[i]:=s[i-1]+Random(3)
            End;
            For i:=1 to ElemSzam Div 10 do
                Begin
                    j:=Random(ElemSzam);
                    Repeat
                        k:=Random(ElemSzam)

```

```

    Until k<>j;
    seged:=s[j]; s[j]:=s[k]; s[k]:=seged;
End;
r:=s;
TombKiir('Föltöltés majdnem rendezve');
Varakozas
End; {FeltoltMajdnemRendezve}
{
    Menüdeklaráció vége
}

Procedure TombKiir(cim: String);
    Var
        i: Integer;
    Begin
        Window(1,1,80,25); {Az aktív ablakterület}
        GotoXY(((80-length(cim) Div 2)+2,1);
        HighVideo; Write(cim); Lowvideo;
        Window(2,2,78,23); {Az aktív ablakterület}
        For i:=1 to ElemSzam do
            Begin
                NormVideo; Write(i:5,':'); HighVideo; Write(r[i]:5);
            End;
        End; {TombKiir}

Procedure Varakozas;
    Var
        c: Char;
    Begin
        Window(1,1,80,25); {Az aktív ablakterület}
        Repeat
            GotoXY(40,24); Write('|'); Delay(100); GotoXY(40,24); Write('/'); Delay
(100);
            GotoXY(40,24); Write('-'); Delay(100); GotoXY(40,24); Write('\'); Delay
(100);
        Until KeyPressed; c:=ReadKey;
        ClrScr;
    End; {Várakozás}

Procedure Ures; {Az üres menüpontokhoz tartozó tevékenység}
    Begin
    End; {üres}

Procedure Vege;
    Begin
        AblakRajzolas(1,1,24,79);
        GotoXY(38,12);
        WriteLn('Vége');
        GotoXY(1,24);
        Delay(1000);
        ClrScr;
    End; {Vége}

Procedure AblakRajzolas(bfx,bfy,hossz,szel: Byte);
    Var
        i: Byte;
    Begin
        Window(bfx,bfy,bfx+szel,bfy+hossz);
        ClrScr;
        For i:=1 to szel do
            Begin
                GotoXY(i,1); Write('-'); GotoXY(i,hossz); Write('-');
            End;
        End;
    End;

```



```

For i:=2 to hossz do
Begin
  GotoXY(1,i); Write('-'); GotoXY(szel,i); Write('-');
End;
End; {AblakRajzolás}

Procedure Menuzes(m: Menu);
Var
  melyik,
  szel   : Byte;

Procedure Help (a : Word);
Begin
  Ablakrajzolas(30,10,13,50);
  GotoXY(5,5);
  Write(a);
  Varakozas;
  Window(m.bfx,m.bfy,m.bfx+szel+2,m.bfy+m.db+2);
End;

Procedure MenuKiiras(m: Menu);
Var
  i: Byte;

Procedure KepernyoKimentes;
Begin

End; {KepernyoKimentes}

Procedure KepernyoVisszamentes;
Begin

End; {KepernyoVisszamentes}

Function MaxSzel(m: Menu): Byte;
Var
  i,j: Byte;
Begin
  j:=Length(m.cim);
  For i:=1 to m.db do
  Begin
    If j<Length(m.am[i].kerdes) then j:=Length(m.am[i].kerdes);
  End;
  MaxSzel:=j
End; {MaxSzel}

Begin {MenüKiírás}
  KepernyoKimentes; {Verembe!!!}
  szel:=MaxSzel(m)+1;
  Ablakrajzolas(m.bfx,m.bfy,m.db+2,szel+2);
  GotoXY(((szel-length(m.cim)) Div 2)+2,1);
  HighVideo; Write(m.cim); Lowvideo;
  For i:=1 to m.db do
  Begin
    GotoXY(2,i+1); Write(m.am[i].kerdes);
    GotoXY(m.am[i].sorsz+1,i+1);
    HighVideo; Write(m.am[i].kerdes[m.am[i].sorsz]); Lowvideo;
  End;
End; {MenüKiírás}

Function MenuValasztas(m: Menu): Byte;
Var
  c   : Char;

```

```

    eleg : Boolean;
    i     : Integer;
    betuk: Set of Char;
Begin {MenüVálasztás}
    betuk:=[];
    For i:=1 to m.db do
    Begin
        c:=Ucase(m.am[i].kerdes[m.am[i].sorsz]);
        betuk:=betuk+[c];
    End;
    i:=1;
    Highvideo; GotoXY(2,i+1); Write(m.am[i].kerdes); Lowvideo;
    eleg:=False;
    Repeat
        GotoXY(1,1);
        c:=Readkey;
        Case c of
            #0 : Begin
                c:=Readkey;
                Case c of
                    #80 : Begin
                        GotoXY(2,i+1); Write(m.am[i].kerdes);
                        GotoXY(m.am[i].sorsz+1,i+1);
                        HighVideo;
                        Write(m.am[i].kerdes[m.am[i].sorsz]);
                        Lowvideo;
                        i:=i+1;
                        If i>m.db then i:=1;
                        Highvideo;
                        GotoXY(2,i+1); Write(m.am[i].kerdes);
                        Lowvideo;
                        c:=#0; {speciális jel}
                    End;
                    #72 : Begin
                        GotoXY(2,i+1); Write(m.am[i].kerdes);
                        GotoXY(m.am[i].sorsz+1,i+1);
                        HighVideo;
                        Write(m.am[i].kerdes[m.am[i].sorsz]);
                        Lowvideo;
                        i:=i-1;
                        If i=0 then i:=m.db;
                        Highvideo;
                        GotoXY(2,i+1); Write(m.am[i].kerdes);
                        Lowvideo;
                        c:=#0; {speciális jel}
                    End;
                    #59 : help(m.am[i].segit);
                End; {case}
            End;
            #13 : eleg:=true;
            #27 : Begin
                i:=m.db; eleg:=True;
            End;
        End; {case};
        If not eleg and (Ucase(c) in betuk) then
        Begin
            i:=1;
            While Ucase(c)<>Ucase(m.am[i].kerdes[m.am[i].sorsz]) do Inc(i);
            eleg:=True;
        End;
    Until eleg;
    MenuValasztas:=i;
End; {MenüVálasztás}

```

```
Begin {Menüzés}
  ClrScr;
  Repeat
    MenuKiiras(m);
    melyik:=MenuValasztas(m);
    m.am[melyik].tevek;
  Until melyik=m.db;
End; {Menüzés}
{A próbához, mint programhoz:
Begin
  Menuzes (FoMenu);
End.
}
```