

18 Dos unit, rendezési algoritmusok

- dos unit
- rendezési algoritmusok

18.1 Dos unit

A dos unit segítségével többek között lekérhetjük az aktuális dátumot és időt. A dátumot a **getdate** eljárás segítségével kérhetjük le. A parancsnak négy paramétere van. Ebbe a négy paraméterként megadott változóba adja vissza az eljárás az aktuális évet, hónapot, napot és azt, hogy a hét melyik napja van éppen (0-vasárnap, 1-hétfő, ...).

Hasonlóan működik az idő lekérésére is a **gettime** eljárás. Paraméterként itt is négy változót kell megadnunk, melyekben visszakapjuk az aktuális órát, percet, másodpercet és századmásodpercet. Ennek az eljárásnak a segítségével fogjuk később lemérni az egyes rendezési algoritmusok futásának időtartamát.

```

program Pelda59;
uses dos crt;
var d1,d2,d3,d4,t1,t2,t3,t4:word;
begin
clrscr;
getdate(d1,d2,d3,d4);
gettime(t1,t2,t3,t4);
write('Mai datum: ',d1,'. ',d2,'. ',d3,'. ');
case d4 of
  0: writeln('vasarnap');
  1: writeln('hetfo');
  2: writeln('kedd');
  3: writeln('szerda');
  4: writeln('csutortok');
  5: writeln('pentek');
  6: writeln('szombat');
end;
writeln('Ido: ',t1,':',t2,':',t3,':',t4);
readln;
end.

```

18.2 Rendezési algoritmusok

Adott egy tömb, melyben véletlenszerű számok vannak. Feladatunk, hogy rendezzük ezeket a számokat növekvő sorrendbe különböző algoritmusok segítségével. A rendezés idejét minden esetben mérjük le századmásodpercekben.

Egyszerű rendezés (SimpleSort)

Ez a programilag legegyszerűbb rendezési algoritmus. Mindegyik elemet összehasonlítjuk az összes utána következő elemmel két egymásba ágyazott ciklus segítségével. Ha az éppen összehasonlított két elem nem jó sorrendben van, akkor azokat rögtön ki is cseréljük.

Rendezés a legkisebb elem kiválasztásával (MinSort, SelectSort)

A módszer lényege, hogy az i . helyre ($i = 1, 2, 3, \dots$) sorban kiválasztjuk az i -...elemekszama helyen levők közül a legkisebbet.

Rendezés a szomszédos elemek cseréjével (BubbleSort)

A módszer lényege, hogy egyesével végignézzük a szomszédos elemeket, és ha előbb van a nagyobb, megcseréljük azokat. Így biztosan a legnagyobb helyre kerül a legnagyobb elem ("a buborék felszáll"). Ugyanezt sorban végrehajtjuk az utolsó előtti elemig stb., egészen az elsőig. Ha közben kiderül, hogy a tömb rendezett, az eljárást befejezzük.

Rendezés beszúrásos módszerrel (InsertSort)

A módszer hasonlít a kártyák kézben való rendezéséhez. Mindig vesszük a következő elemet, s azt a megfelelő helyre beszúrjuk úgy, hogy a többi, már rendezett elemet eggyel feljebb toljuk.

Feladat: Mindegyik rendezés pontos algoritmusát megtalálható az alábbi programban:

```

program Pelda60;
uses crt, dos;
const elemekszama = 10000;
var a,b:array [1..elemekszama] of integer;
    ido:longint;
    i:integer;

procedure kiiras;
var i:integer;
begin
  for i:=1 to elemekszama do write(a[i],' ');
  writeln;
  writeln;
end;

procedure stopperbe;
var t1,t2,t3,t4:word;
begin
  gettime(t1,t2,t3,t4);
  ido:=t4+100*(t3+60*(t2+60*t1));
end;

procedure stopperki;
var t1,t2,t3,t4:word;
begin
  gettime(t1,t2,t3,t4);
  ido:=t4+100*(t3+60*(t2+60*t1))-ido;
end;

{ egyszeru rendezes }
procedure simplesort;
var i,j,x:integer;
begin
  for i:=1 to elemekszama-1 do
    for j:=i+1 to elemekszama do
      if a[i]>a[j] then begin
        x:=a[i];
        a[i]:=a[j];
        a[j]:=x;
      end;
end;

{ rendezes a legkisebb elem kiválasztásával }
procedure minsort;
var i,j,min,x:integer;
begin
  for i:=1 to elemekszama-1 do
    begin
      min:=i;
      for j:=i+1 to elemekszama do
        if a[j]<a[min] then min:=j;
      x:=a[i];
      a[i]:=a[min];
      a[min]:=x;
    end;
end;

```

```

procedure bubblesort; { rendezes a szomszedos elemek cserejevel }
var i,j,x:integer;
    ok:boolean;
begin
    i:=elemekszama-1;
    ok:=false;
    while (i>=1) and (not ok) do
        begin
            ok:=true;
            for j:=1 to i do
                if a[j]>a[j+1] then begin
                    x:=a[j];
                    a[j]:=a[j+1];
                    a[j+1]:=x;
                    ok:=false;
                end;
            dec(i);
        end;
end;
procedure insertsort; { rendezes beszurasos modszerral }
var i,j,x,ment:integer;
begin
    for i:=2 to elemekszama do
        begin
            if a[i]<a[i-1] then begin
                ment:=a[i];
                j:=i;
                repeat
                    dec(j);
                    a[j+1]:=a[j];
                until (j=1) or (a[j-1]<=ment);
                a[j]:=ment;
            end;
        end;
end;
begin { foprogram }
    clrscr;
    writeln(elemekszama,' drb. elem rendezese:');
    writeln;
    randomize;
    for i:=1 to elemekszama do b[i]:=random(64000)-32000;
    { mindig a-t fogjuk rendezni, b-ben meghagyjuk a kigeneralt szamokat}
    a:=b;
    stopperbe;
    simplesort; { rendezes }
    stopperki;
    writeln('SimpleSort: ',ido,' szazadmasodperc'); { kiiras }
    a:=b;
    stopperbe;
    minsort; { rendezes }
    stopperki;
    writeln('MinSort: ',ido,' szazadmasodperc'); { kiiras }
    a:=b;
    stopperbe;
    bubblesort; { rendezes }
    stopperki;
    writeln('BubbleSort: ',ido,' szazadmasodperc'); { kiiras }
    a:=b;
    stopperbe;
    insertsort; { rendezes }
    stopperki;
    writeln('InsertSort: ',ido,' szazadmasodperc'); { kiiras }
end.

```