

Jednorozmerné pole – triedenie

TRIEDENIE VÝMENOU (preublávaním)

V usporiadanom poli vieme rýchlejšie nájsť hľadanú položku. Preto je dôležité mať položky v poli usporiadané, teda zotriedené. Triediť môžeme rôznymi spôsobmi. Každý spôsob má svoje výhody i nevýhody.

Princíp triedenia výmenou (preublávaním) spočíva v cyklickom porovnávaní dvoch susedných položiek. Ak potrebujeme pole zotriediť vzostupne a ak prvé porovnávané číslo je väčšie ako druhé, musíme čísla vymeniť. Ak prejdeme celým polom toľkokrát koľko je položiek v poli, pole bude utriedené. Malé čísla ako bublinky sa budú hromadiť na jednej strane a veľké čísla na druhej.

Toto triedenie je časovo veľmi náročné a neefektívne, ale zato je principiálne najjednoduchšie.

V Pascale, kde potrebujeme zoradiť údaje v poli a, by sme to robili takto:

```
for j:=1 to n do
  for i:=1 to n-1 do
    begin
      if a[i]>a[i+1] then
        begin
          p:=a[i];
          a[i]:=a[i+1];
          a[i+1]:=p;
        end;
    end;
end;
```

TRIEDENIE VÝBEROM

Princíp triedenia výberom je nasledovný: na začiatku je celé pole nezotriedené. Prvú položku poľa v nezotriedenej časti postupne porovnáваме s ostatnými položkami v tejto časti a ak nájdeme položku menšiu ako je prvá položka, tak ich navzájom vymeníme. Po porovnaní prvej položky so všetkými ostatnými je na prvej pozícii najmenšia položka, preto túto položku zaradíme do zotriedenej časti poľa a nezotriedenú časť poľa zmenšíme o prvú položku. Celý proces opakujeme dovtedy, kým nebudú všetky položky poľa patriť do zotriedenej časti a teda nezotriedená časť poľa bude prázdna.

Toto triedenie nie je časovo veľmi náročné a principiálne je pomerne jednoduché.

V Pascale, kde potrebujeme zoradiť údaje v poli a, by sme to robili takto:

```
for j:=1 to n-1 do
  for i:=j+1 to n do
```

```

begin
  if a[i]<a[j] then
    begin
      p:=a[i];
      a[i]:=a[j];
      a[j]:=p;
    end;
  end;
end;

```

TRIEDENIE VSÚVANÍM

Princíp triedenia vsúvaním je nasledovný: na začiatku rozdelíme pole na dve časti do prvej – zotriedenej bude patriť prvá položka, ostatné položky budú patriť do nezotriedenej časti. Zoberieme prvú položku z nezotriedenej časti, uložíme ju do pomocnej premennej a hľadáme pre ňu miesto v zotriedenej časti, tak aby táto časť bola po zaradení tejto položky i naďalej usporiadaná. Po nájdení miesta vyposúvame položky za týmto miestom tak, že na miesto prvej položky v nezotriedenej časti uložíme poslednú položku zo zotriedenej časti, na jej miesto príde predchádzajúca a tento proces robíme dovtedy, kým nedôjdeme k miestu, kde treba uložiť položku z pomocnej premennej, nakoniec ju na uvoľnené miesto v poli uložíme. Zväčšíme počet položiek usporiadanej časti poľa o jedna a počet položiek v neusporiadanej časti zmenšíme o jedna. Celý proces opakujeme, kým nebudú všetky položky poľa patriť do zotriedenej časti a teda nezotriedená časť poľa bude prázdna.

Toto triedenie nie je časovo veľmi náročné a principiálne je oproti predošlému náročnejšie.

V Pascale, kde potrebujeme zoradiť údaje v poli a, by sme to robili takto:

```

for k:=2 to n do
  begin
    {vyberieme prvý prvok z neusporiadanej casti, ktory
    treba zaradit}
    x:=a[k];
    i:=1;

    {hľadame miesto pre tento prvok}
    while x>a[i] do inc(i);
    j:=k-1;

    {posuvame prvky tak, aby sme vytvorili miesto pre
    prvok}
    while j>=i do
      begin
        a[j+1]:=a[j];
        dec(j);
      end;
    end;
  end;
end;

```

```

end;

{zaradime prvok na spravne miesto}
a[i]:=x;
end;

```

REKURZÍVNE TRIEDENIE

Princíp rekurzívneho triedenia je nasledovný: Najprv budeme deliť pole na menšie úseky, kým sa úsek rozdeliť bude dať a potom úseky spojíme do jedného celku tak, aby údaje v spojených úsekoch boli usporiadané. Napríklad usporiadajme pole s prvkami 5, -3, 8, 0, -7. Delíme pole na dva úseky: 5, -3, 8 a 0, -7. Prvý úsek rozdelíme na dva úseky: 5, -3 a 8. Prvý úsek rozdelíme na úsek 5 a -3. Ďalej sa už prvý úsek rozkladať nedá. Ani druhý úsek sa už rozkladať nedá, tak ich zlúčime do pomocného poľa b, ktorého obsah potom prepíšeme na príslušné miesta v poli a. Po zlúčení bude obsah poľa a vyzerá takto: -3, 5, 8, 0, -7. Úsek s číslom 8 sa ďalej rozkladať nedá, preto ho zlúčime s usporiadaným úsekom -3, 5. Po zlúčení bude obsah poľa a vyzerá takto: -3, 5, 8, 0, -7. Keďže prvé tri položky poľa a sú usporiadané, pokračujeme v delení druhého úseku: 0, -7. Rozdelíme ho na úsek 0 a -7. Keďže obidva úseky sa už ďalej rozkladať nedajú, zlúčime ich. Po zlúčení vyzerá obsah poľa a takto: -3, 5, 8, -7, 0. Nakoniec zlúčime dokopy úseky -3, 5, 8 a -7, 0 takto: porovnáme prvé položky z oboch úsekov a do poľa b zapíšeme tú menšiu teda -7, potom porovnáme -3 s 0, do poľa b sa na druhé miesto dostane -3, ďalej porovnáme 5 s 0, do poľa b sa na tretiu pozíciu dostane 0. Keďže sme už vyčerpali všetky položky z druhého úseku dopíšeme do poľa b zvyšné položky z prvej časti, teda 5, 8. Po prepise obsahu poľa b do poľa a, vyzerá jeho obsah takto: -7, -3, 0, 5, 8 a je usporiadaný!

Toto triedenie je časovo najoptimálnejšie, ale principiálne je pomerne zložité.

V Pascale, kde potrebujeme zoradiť údaje v poli a, by sme to robili takto:

V hlavnom programe by sme volali procedúru tried
tried(1,n);

Procedúra tried vyzerá takto:

```

procedure tried(m,n:integer);
var s:integer;
begin
    if m<n then
        begin    {delenie pola na dva useky}
            s:=(n+m) div 2;
            tried(m,s);
            tried(s+1,n);
            zluc(m,s,n);    {zlucenie usekov do usporiadaneho
celku}
        end;

```

end;

Najzložitejšou je procedúra zluc, ktorá vyzerá takto:

```
procedure zluc(m,s,n:integer);
var b:array[1..20] of real;
    j,k,l:integer;
begin
    j:=m;
    k:=s+1;
    l:=m;

    {zapis mensich prvkov do pola b z jednotlivych usekov}
    while (j<=s) and (k<=n) do
        begin
            if a[j]<=a[k] then
                begin
                    b[l]:=a[j];
                    inc(j);
                end
            else
                begin
                    b[l]:=a[k];
                    inc(k);
                end;
            inc(l);
        end;

    {dopisanie prvkov z prveho useku do pola b}
    while j<=s do
        begin
            b[l]:=a[j];
            inc(j);
            inc(l);
        end;

    {dopisanie prvkov z druheho useku do pola b}
    while k<=n do
        begin
            b[l]:=a[k];
            inc(k);
            inc(l);
        end;

    {prepis upravovanej casti pola b do pola a}
    for j:=m to n do a[j]:=b[j];
end;
```