

Mergesort

Programování 2
(NMIN102)

RNDr. Michal Žemlička, Ph.D.

Mergesort

- třídící algoritmus využívající slévání
- potřebuje pomocný prostor velikosti tříděných dat + pár proměnných
- pracuje v čase $O(n \log n)$

Slévání

- proces spojující několik (dvě nebo více) monotónních posloupností do jedné
- všechny posloupnosti dodržují stejnou monotonii (jsou buď všechny neklesající nebo nerostoucí; totéž pak platí pro posloupnost výslednou)
- v lineárním čase (každý prvek jednou přečteme a jednou zapíšeme)

Slévání (2)

- je možné je provádět na datech v souborech i v paměti
- monotónní posloupnosti mohou být pevné nebo proměnné velikosti
- můžeme mít uložených více monotónních posloupností na jednom médiu (v jednom poli či souboru)

Slévání – Idea

- z prvků aktuálně načtených z jednotlivých posloupností vybereme nejmenší;
- zapíšeme jej na výstup;
- z dané posloupnosti načteme nový prvek.

Hlavička

CONST

MaxN = 10000; { limit počtu prvků ve tříděném poli }

TYPE

PRVEK = Integer;

ROZSAH = 1..MaxN;

POLE = ARRAY [ROZSAH] OF PRVEK;

VAR

a : POLE;

N : Integer; { skutečný počet prvků v poli }

Výpis

```
PROCEDURE vypis(CONST a:Pole);  
  VAR  
    i : ROZSAH;  
  BEGIN  
    FOR i:=1 TO N-1 DO Write(a[i],', ', ');  
    WriteLn(a[N],'.');  
  END;
```

Vypíše výsledek; ale může se hodit i pro ladící tisky.

Slévání dvou posloupností

```
PROCEDURE Sluc(VAR Zdroj,Cil:Pole; od1,do1,od2,do2,zac,kon:Rozsah);
VAR
    i1, i2, j : Integer; { ukazovatka }
    x1, x2 : Prvek;
PROCEDURE Z_Prvni; { cteme z prvni posloupnosti }
BEGIN
    Cil[j] := x1; j := j + 1; { zapiseme }
    i1 := i1 + 1; { dalsi prvek na prvni vstupu }
    IF i1 <= do1 THEN x1 := Zdroj[i1];
END;
```

Slévání dvou posloupností (pokrač.)

```
PROCEDURE Z_Druhe; { cteme z druhe posloupnosti }
BEGIN
  Cil[j] := x2; j := j + 1; { zapiseme }
  i2 := i2 + 1; { dalsi prvek na druhem vstupu }
  IF i2 <= do2 THEN x2 := Zdroj[i2];
END;

BEGIN
  x1:=Zdroj[od1]; x2:=Zdroj[od2]; i1:=od1; i2:=od2; j:=zac;
  WHILE (i1<=do1)AND(i2<=do2) DO
    IF x1 < x2 THEN Z_Prveni
      ELSE Z_Druhe;
  WHILE i1<=do1 DO Z_Prveni;
  WHILE i2<=do2 DO Z_Druhe;
END;
```

Slévání všech posloupností v poli

```
PROCEDURE Faze(VAR Zdroj, Cil:Pole; Krok,Max:Integer);
VAR
  i, dvojkrok, s{tart} : Integer;
BEGIN
  dvojkrok := 2*Krok; s:=1;
  FOR i:=1 TO Max DIV dvojkrok DO { pro jednotlivá úplná slucování }
    BEGIN
      Sluc(Zdroj,Cil,s,s+krok-1,s+krok,s+dvojkrok-1,s,s+dvojkrok-1);
      s := s + dvojkrok;
    END;
  IF start <= Max-krok THEN { slucujeme menší kus }
    Sluc(Zdroj,Cil,s,s+krok-1,s+krok,Max,s,Max)
  ELSE IF s<=Max THEN { zbyva jen usporadany kousek }
    FOR i:=s TO Max DO cil[i]:=zdroj[i];
  END;
END;
```

Mergesort

```
PROCEDURE MergeSort(VAR x:Pole);
VAR
  xy : Boolean; { slevame z pole x do pole y? }
  y : Pole;
  krok, i : Integer;
BEGIN
  Krok:=1; xy:=true;
  WHILE Krok<=PocetPrvku DO
    BEGIN
      IF xy THEN Faze(x,y,krok,PocetPrvku)
        ELSE Faze(y,x,krok,PocetPrvku);
      xy := NOT xy;   Krok := 2 * Krok;
    END;
  IF NOT xy THEN { pokud je vysledek v y, prekopiruj jej do x }
    x:=y;
  END;
```