

Heapsort

Programování 2 – cvičení  
(NMIN102)

RNDr. Michal Žemlička, Ph.D.

# Heapsort

- třídící algoritmus využívající haldu
- vystačí s polem, co třídí, a pár proměnnými
- pracuje v čase  $O(n \log n)$

# Halda (heap)

- datová struktura třídící algoritmus využívající haldu
- určeno pro snadné nalazení extrému (maximum – max-halda, minimum minhalda; liší se jen uspořádáním)
- dá se vytvořit v lineárním čase
- struktura:  $d$ -ární strom (známe pro  $d = 2$ )
- extrém dostupný v  $O(1)$

# Hlavička

CONST

N = 7; { pocet prvku v tridenem poli }

d = 2; { arita haldy }

TYPE

PRVEK = Integer;

ROZSAH = 1..N;

POLE = ARRAY [ROZSAH] OF PRVEK;

VAR

a : POLE;

# Výpis

```
PROCEDURE vypis;  
  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.

# Výběr kandidáta na případné prohození

```
FUNCTION vyberKandidata(boss:ROZSAH):ROZSAH;  
  { vraci index prvku, ktery by mohl byt prohozen s nadrazenym }  
  VAR  
    i, j, m, k : ROZSAH;  
    x : PRVEK;  
  BEGIN  
    j:=d*(boss-1)+2;  
    IF j+d-1 > Posledni THEN m:=Posledni ELSE m:=j+d-1;  
    x := a[j]; k := j;  
    FOR i:=j+1 TO m DO  
      IF a[i]>x THEN BEGIN x:=a[i]; k:=i END;  
    vyberKandidata := k;  
  END;
```

pro  $d = 2$  stačí pouhé porovnání

# Zařazení prvku do haldy

```
PROCEDURE zaradPrvek(i:ROZSAH);  
  VAR  
    j : ROZSAH;  
    x : PRVEK;  
  BEGIN  
    IF d*i-(d-2) > Posledni THEN Exit; { uz jsme v listu }  
    j:=vyberKandidata(i);  
    IF a[i]<a[j] THEN { pouze pokud je neco v neporadku }  
      BEGIN  
        x:=a[i]; a[i]:=a[j]; a[j]:=x; { prohozeni }  
        zaradPrvek(j);  
      END;  
    END;  
  END;
```

# Vytvoření haldy

```
PROCEDURE vytvorHaldu;  
  VAR  
    i : ROZSAH;  
  BEGIN  
    FOR i:=(N+d-2) DIV d DOWNTO 1 DO zaradPrvek(i);  
  END;
```

# Třídíme

```
PROCEDURE tridime;
  VAR
    i : ROZSAH;
    x : PRVEK;
  BEGIN
    FOR i:=N DOWNTO 2 DO
      BEGIN
        Posledni := i; { hledame, kam az se pracuje s daty }
        zaradPrvek(1);
        { prohozeni prvnioho s aktualne poslednim }
        x:=a[i]; a[i]:=a[1]; a[1]:=x;
      END
    END;
END;
```