

Řazení údajů

Programovací techniky

doc. Ing. Jiří Rybička, Dr.
Ústav informatiky
PEF MENDELU v Brně
rybicka@mendelu.cz

- **Řazení** = uspořádání údajů podle velikosti

- **Řazení** = uspořádání údajů podle velikosti
- Existuje relace (úplného) **uspořádání** na množině dat

- **Řazení** = uspořádání údajů podle velikosti
- Existuje relace (úplného) **uspořádání** na množině dat
- **Třídění** = rozdělení množiny dat na části (rozklad množiny)

- **Řazení** = uspořádání údajů podle velikosti
- Existuje relace (úplného) **uspořádání** na množině dat
- **Třídění** = rozdělení množiny dat na části (rozklad množiny)
- Existuje relace **ekvivalence**

- Přímé metody = realizace základního principu

- Přímé metody = realizace základního principu
- Modifikované metody = „zlepšení“ principu

- Přímé metody = realizace základního principu
- Modifikované metody = „zlepšení“ principu
- přímé řazení / nepřímé řazení =
přeskupují/nepřeskupují se řazené údaje
(databázové indexové soubory)

- Přímé metody = realizace základního principu
- Modifikované metody = „zlepšení“ principu
- přímé řazení / nepřímé řazení =
přeskupují/nepřeskupují se řazené údaje
(databázové indexové soubory)
- Sekvenční metody = pracují pouze se sousedními prvky

- Přímé metody = realizace základního principu
- Modifikované metody = „zlepšení“ principu
- přímé řazení / nepřímé řazení =
přeskupují/nepřeskupují se řazené údaje
(databázové indexové soubory)
- Sekvenční metody = pracují pouze se sousedními prvky
- Přímopřístupové metody = potřebují přístup k libovolnému prvku struktury

- Přímé metody = realizace základního principu
- Modifikované metody = „zlepšení“ principu
- přímé řazení / nepřímé řazení =
přeskupují/nepřeskupují se řazené údaje
(databázové indexové soubory)
- Sekvenční metody = pracují pouze se sousedními prvky
- Přímopřístupové metody = potřebují přístup k libovolnému prvku struktury
- Vnitřní řazení = v operační paměti

- Přímé metody = realizace základního principu
- Modifikované metody = „zlepšení“ principu
- přímé řazení / nepřímé řazení =
přeskupují/nepřeskupují se řazené údaje
(databázové indexové soubory)
- Sekvenční metody = pracují pouze se sousedními prvky
- Přímopřístupové metody = potřebují přístup k libovolnému prvku struktury
- Vnitřní řazení = v operační paměti
- Vnější řazení = přímo v souborech na disku

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- výběrové – vyhledávání význačného prvku (extrému)

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- výběrové – vyhledávání význačného prvku (extrému)
- vkládací – do seřazené struktury se vkládá na vhodné místo nový prvek

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- výběrové – vyhledávání význačného prvku (extrému)
- vkládací – do seřazené struktury se vkládá na vhodné místo nový prvek
- rozdělovací – struktura se rozdělí na části a v nich se provádí řazení

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- výběrové – vyhledávání význačného prvku (extrému)
- vkládací – do seřazené struktury se vkládá na vhodné místo nový prvek
- rozdělovací – struktura se rozdělí na části a v nich se provádí řazení
- slučovací – z částí se skládá (částečně/úplně) uspořádaný celek

- výběrové – vyhledávání význačného prvku (extrému)
- vkládací – do seřazené struktury se vkládá na vhodné místo nový prvek
- rozdělovací – struktura se rozdělí na části a v nich se provádí řazení
- slučovací – z částí se skládá (částečně/úplně) uspořádaný celek
- Výběr je singulární případ rozdělování (1 prvek $\times$ zbytek)

- výběrové – vyhledávání význačného prvku (extrému)
- vkládací – do seřazené struktury se vkládá na vhodné místo nový prvek
- rozdělovací – struktura se rozdělí na části a v nich se provádí řazení
- slučovací – z částí se skládá (částečně/úplně) uspořádaný celek
- Výběr je singulární případ rozdělování (1 prvek $\times$ zbytek)
- Vkládání je singulární případ slučování (1 prvek $\times$ zbytek)

- Přirozené: jsou schopny využít částečně seřazených dat

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)
- Podle časové složitosti:

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)
- Podle časové složitosti:
 - lineární (teoretické minimum pro speciální případy),

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)
- Podle časové složitosti:
 - lineární (teoretické minimum pro speciální případy),
 - lineárně logaritmické (dobré univerzální metody),

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)
- Podle časové složitosti:
 - lineární (teoretické minimum pro speciální případy),
 - lineárně logaritmické (dobré univerzální metody),
 - kvadratické (rekvizity začátečníků),

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)
- Podle časové složitosti:
 - lineární (teoretické minimum pro speciální případy),
 - lineárně logaritmické (dobré univerzální metody),
 - kvadratické (rekvizity začátečníků),
 - horší (zmetky začátečníků)

- Přirozené: jsou schopny využít částečně seřazených dat
- Stabilní: data se stejnými klíči se nepřeskupují
- In situ: pracují pouze s prostorem dat ($O(N) = k \cdot N$)
- Podle časové složitosti:
 - lineární (teoretické minimum pro speciální případy),
 - lineárně logaritmické (dobré univerzální metody),
 - kvadratické (rekvizity začátečníků),
 - horší (zmetky začátečníků)
- Ve všech dalších příkladech: Pole p , obsazeno prvních n indexů, data jsou skalární (úplně porovnatelná).

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight
Selection)

Bublínové řazení (Bubble
Sort)

Řazení hromadou (Heap
Sort)

Vkládací metody

Rozdělovací
metody

Slučovací metody

- Princip: výběr extrému a jeho uložení do cílového pole

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Princip: výběr extrému a jeho uložení do cílového pole
- Varianta in-situ: pole má dvě části

- Princip: výběr extrému a jeho uložení do cílového pole
- Varianta in-situ: pole má dvě části
- Obecná procedura pro záměnu prvků:

```
void Zamena(Pole P, Indexy A, Indexy B){  
    Slozka Pom;  
    if (A!=B) {  
        Pom=P[A];  
        P[A]=P[B];  
        P[B]=Pom;  
    }  
}
```

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Funkce pro nalezení extrému mezi zadanými indexy:

```
Indexy Extrem(Pole P, Indexy A, Indexy B){  
    Indexy V = A, I;  
    for (I=A+1; I<=B; I++)  
        if (P[I]<P[V]) V=I;  
    return V;  
}
```

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Funkce pro nalezení extrému mezi zadanými indexy:

```
Indexy Extrem(Pole P, Indexy A, Indexy B){  
    Indexy V = A, I;  
    for (I=A+1; I<=B; I++)  
        if (P[I]<P[V]) V=I;  
    return V;  
}
```

- Vlastní řazení:

```
for (Indexy I=N; I>=2; I--)  
    Zamena(P, Extrem(P, 1, I), I);
```

- Funkce pro nalezení extrému mezi zadanými indexy:

```
Indexy Extrem(Pole P, Indexy A, Indexy B){  
    Indexy V = A, I;  
    for (I=A+1; I<=B; I++)  
        if (P[I]<P[V]) V=I;  
    return V;  
}
```

- Vlastní řazení:

```
for (Indexy I=N; I>=2; I--)  
    Zamena(P, Extrem(P, 1, I), I);
```

- Časová složitost kvadratická ($O(N) = k \cdot N^2$)

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Populární u začátečníků

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Populární u začátečníků
- Zajímavá varianta: s detekcí uspořádání – přirozená

```
int J=0;
bool Jeste=true;
while (Jeste) {
    Jeste=false;
    for (int I=0; I<N-J-1; I++)
        if (P[I]>P[I+1]) {
            Zamena(P, I, I+1);
            Jeste=true;
        }
    J++;
}
```

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Populární u začátečníků
- Zajímavá varianta: s detekcí uspořádání – přirozená

```
int J=0;
bool Jeste=true;
while (Jeste) {
    Jeste=false;
    for (int I=0; I<N-J-1; I++)
        if (P[I]>P[I+1]) {
            Zamena(P, I, I+1);
            Jeste=true;
        }
    J++;
}
```

- Přirozená, **sekvenční**, stabilní

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Populární u začátečníků
- Zajímavá varianta: s detekcí uspořádání – přirozená

```
int J=0;
bool Jeste=true;
while (Jeste) {
    Jeste=false;
    for (int I=0; I<N-J-1; I++)
        if (P[I]>P[I+1]) {
            Zamena(P, I, I+1);
            Jeste=true;
        }
    J++;
}
```

- Přirozená, **sekvenční**, stabilní
- Implementační jednoduchost

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Populární u začátečníků
- Zajímavá varianta: s detekcí uspořádání – přirozená

```
int J=0;
bool Jeste=true;
while (Jeste) {
    Jeste=false;
    for (int I=0; I<N-J-1; I++)
        if (P[I]>P[I+1]) {
            Zamena(P, I, I+1);
            Jeste=true;
        }
    J++;
}
```

- Přirozená, **sekvenční**, stabilní
- Implementační jednoduchost
- Časová složitost kvadratická

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Pojem hromada – uspořádaný strom, kde otec je větší než libovolný syn, mezi syny není definováno uspořádání

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Pojem hromada – uspořádaný strom, kde otec je větší než libovolný syn, mezi syny není definováno uspořádání
- Ustavení hromady – přehození prvků tak, aby platilo pravidlo hromady

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Pojem hromada – uspořádaný strom, kde otec je větší než libovolný syn, mezi syny není definováno uspořádání
- Ustavení hromady – přehození prvků tak, aby platilo pravidlo hromady
- Kořen stromu je extrém

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Pojem hromada – uspořádaný strom, kde otec je větší než libovolný syn, mezi syny není definováno uspořádání
- Ustavení hromady – přehození prvků tak, aby platilo pravidlo hromady
- Kořen stromu je extrém
- Strom je reprezentován polem, index levého syna je dvojnásobkem indexu otce, index pravého syna je následníkem levého syna

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Pojem hromada – uspořádaný strom, kde otec je větší než libovolný syn, mezi syny není definováno uspořádání
- Ustavení hromady – přehození prvků tak, aby platilo pravidlo hromady
- Kořen stromu je extrém
- Strom je reprezentován polem, index levého syna je dvojnásobkem indexu otce, index pravého syna je následníkem levého syna
- Velmi efektivní metoda

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Vlastní řazení má dva kroky:

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Vlastní řazení má dva kroky:

- ustavení hromady

- | `for (int I=(N-1)/2; I>=0; I--) Sift(P, I, N-1);`

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Vlastní řazení má dva kroky:

- ustavení hromady

- | `for (int I=(N-1)/2; I>=0; I--) Sift(P, I, N-1);`

- postupné odebírání kořene a ustavení hromady

- | `for (int I=N-1; I>=1; I--){`
 - `Zamena(P, 0, I);`
 - `Sift(P, 0, I-1);`
 - `}`

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Klíčová procedura Sift

```
void Sift(int P[], Indexy L, Indexy R){
    Indexy I = L, J = I*2;
    Prvek Pom = P[I];
    bool Jeste = J<=R;
    while (Jeste) {
        if (J<R and P[J]<P[J+1]) J++;
        if (P[I]<P[J]) {
            P[I]=P[J];
            I=J; J=I*2;
            Jeste = J<=R
        } else Jeste=false;
        P[I]=Pom;
    }
}
```

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Nepřirozená (tj. zcela nezávislá na stupni seřazení dat), nestabilní, in situ

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Nepřirozená (tj. zcela nezávislá na stupni seřazení dat), nestabilní, in situ
- Časová složitost jedna z nejlepších,
 $O(N) = k \cdot N \cdot \log_2 N$

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublinové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Nepřirozená (tj. zcela nezávislá na stupni seřazení dat), nestabilní, in situ
- Časová složitost jedna z nejlepších, $O(N) = k \cdot N \cdot \log_2 N$
- Univerzální metoda

Základní pojmy

Metody

Výběrové metody

Přímý výběr (Straight Selection)

Bublínové řazení (Bubble Sort)

Řazení hromadou (Heap Sort)

Vkládací metody

Rozdělovací metody

Slučovací metody

- Nepřirozená (tj. zcela nezávislá na stupni seřazení dat), nestabilní, in situ
- Časová složitost jedna z nejlepších,
 $O(N) = k \cdot N \cdot \log_2 N$
- Univerzální metoda
- Implementační konstanty bývají o něco horší než u QuickSort

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- nalezení vhodného místa pro vkládaný prvek

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- nalezení vhodného místa pro vkládaný prvek
- odsun následných prvků

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- nalezení vhodného místa pro vkládaný prvek
- odsun následných prvků
- vložení nového prvku

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- nalezení vhodného místa pro vkládaný prvek
- odsun následných prvků
- vložení nového prvku
- in-situ, nesequenční, stabilní varianta

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- nalezení vhodného místa pro vkládaný prvek
- odsun následných prvků
- vložení nového prvku
- in-situ, nesequenční, stabilní varianta
- časová složitost $k \cdot N^2$

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- nalezení vhodného místa pro vkládaný prvek
- odsun následných prvků
- vložení nového prvku
- in-situ, nesequenční, stabilní varianta
- časová složitost $k \cdot N^2$
- pokud je hledání půlením intervalu, lze zlepšit časovou složitost o jednu třídu na lineárně logaritmickou, $O(N) = k \cdot N \cdot \log_2 N$

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Implementace:

```
int vyhledej(int p[], int odkud, int pokud,
             int prvek){
    int index=odkud;
    while ((index<pokud) and (prvek<p[index]))
        index++;
    return index;
}

void odsun(int p[], int odkud, int kam){
    for (int i=kam; i>odkud; i--) p[i]=p[i-1];
}
```

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Tělo řadicí procedury:

```
int novy, kam;
for (int i=0; i<N; i++) {
    //i ukazuje na konec seřazené části pole
    novy=pole[N-1];
    //bereme vždy poslední v poli jako nový
    kam=vyhledej(pole, 0, i, novy);
    //kam přijde nový prvek?
    odsun(pole,kam,N-1);
    //uděláme místo -> odsuneme do konce pole
    pole[kam]=novy;
    //vložíme na volné místo
}
```

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Vkládání nových prvků do stromové struktury v uspořádání $ls \leq o, o \leq ps$

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Vkládání nových prvků do stromové struktury v uspořádání $ls \leq o, o \leq ps$
- Následné provedení operace inorder

```
void Vloz(UkUzel &S, Prvek D){  
    if (S==NULL) {  
        S = new Uzel;  
        S->Data=D;  
        S->Vlevo=NULL;  
        S->Vpravo=NULL;  
    } else  
        if (D<S->Data) Vloz(S->Vlevo, D);  
        else Vloz(S->Vpravo,D);  
}
```

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight
insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací
metody

Slučovací metody

- Vlastní řazení:

```
UkUzel S = NULL;  
Prvek X;  
while (cin>>X) Vloz(S, X);  
Inorder(S);
```

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight
insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací
metody

Slučovací metody

- Vlastní řazení:

```
UkUzel S = NULL;  
Prvek X;  
while (cin>>X) Vloz(S, X);  
Inorder(S);
```

- časová složitost je $k \cdot N \cdot \log_2 N$, nestabilní,

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Vlastní řazení:

```
UkUzel S = NULL;  
Prvek X;  
while (cin>>X) Vloz(S, X);  
Inorder(S);
```

- časová složitost je $k \cdot N \cdot \log_2 N$, nestabilní,
- nepřírozená (antipřírozená) s hrozícím kolapsem do kvadratické složitosti

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight
insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací
metody

Slučovací metody

- Vlastní řazení:

```
UkUzel S = NULL;  
Prvek X;  
while (cin>>X) Vloz(S, X);  
Inorder(S);
```

- časová složitost je $k \cdot N \cdot \log_2 N$, nestabilní,
- nepřirozená (antipřirozená) s hrozícím kolapsem do kvadratické složitosti
- implementačně jednoduchá, prostorová složitost úměrná rekurzivnímu zanoření $O(S) = k \cdot \log_2 N$

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Ideální časová složitost (lineární)

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Ideální časová složitost (lineární)
- Velké omezení typu dat

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Ideální časová složitost (lineární)
- Velké omezení typu dat
- Implementace: logické pole, celočíselné pole (opakující se hodnoty)

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Ideální časová složitost (lineární)
- Velké omezení typu dat
- Implementace: logické pole, celočíselné pole (opakující se hodnoty)
- Někdy velké prostorové nároky

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Ideální časová složitost (lineární)
- Velké omezení typu dat
- Implementace: logické pole, celočíselné pole (opakující se hodnoty)
- Někdy velké prostorové nároky
- Základní princip: hodnoty nejsou složkami, ale indexy

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

```
typedef unsigned int TypData; //vhodný typ
const TypData XXX = ???; //omezení intervalu
bool M[XXX]; //množina
TypData d;
//vyprázdnění množiny:
for (TypData i=0; i<XXX; i++) M[i]=false;
//naplnění množiny ze vstupu:
while (cin>>d) M[d]=true;
//seřazený výpis prvků množiny
for (TypData i=0; i<XXX; i++)
    if (M[d]) cout<<d<<" ";
```

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight
insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací
metody

Slučovací metody

- Hodnoty libovolného typu převádíme (rozptylováním) do hodnot indexového typu

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Hodnoty libovolného typu převádíme (rozptylováním) do hodnot indexového typu
- V synonymech rozptylování musí existovat uspořádání ve stejném smyslu jako v původních datech

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Hodnoty libovolného typu převádíme (rozptylováním) do hodnot indexového typu
- V synonymech rozptylování musí existovat uspořádání ve stejném smyslu jako v původních datech
- Lze překročit omezení dat při řazení množinou

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Přímé vkládání (straight insert)

Řazení bin. stromem

Řazení množinou

Řazení rozptylováním

Rozdělovací metody

Slučovací metody

- Hodnoty libovolného typu převádíme (rozptylováním) do hodnot indexového typu
- V synonymech rozptylování musí existovat uspořádání ve stejném smyslu jako v původních datech
- Lze překročit omezení dat při řazení množinou
- Efektivita metody je závislá na rozptylovací funkci, jejíž vlastnosti (zachování uspořádání) jsou obvykle náročné na implementaci

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Quick Sort – Rozděl a panuj

Slučovací metody

- Jedna z nejrychlejších metod (ne zcela univerzální)

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Quick Sort – Rozděl a panuj

Slučovací metody

- Jedna z nejrychlejších metod (ne zcela univerzální)
- Rekurzivní zápis vede k větší prostorové složitosti, u nerekurzivního zápisu je nutné implementovat zásobník intervalů

- Jedna z nejrychlejších metod (ne zcela univerzální)
- Rekurzivní zápis vede k větší prostorové složitosti, u nerekurzivního zápisu je nutné implementovat zásobník intervalů
- Základní algoritmus:

```
void quick(int P[], int L, int R){  
    Rozdel;    //Hoare  
    if (L<J) quick(P, L, J);  
    if (I<R) quick(P, I, R);  
}
```

- Kompletní zápis:

```
void quick(int p[], int levy, int pravy){
    int i=levy, j=pravy;
    int pivot=p[(levy+pravy)/2];
    do {
        while (p[i]<pivot) i++;
        while (p[j]>pivot) j--;
        if (i<=j) {
            zamena(p, i, j);
            i++; j--;
        }
    } while (i<=j);
    if (levy<j) quick(p,levy,j);
    if (i<pravy) quick(p,i,pravy);
}
```

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Quick Sort – Rozděl a panuj

Slučovací metody

- Časová složitost: lineárně logaritmická

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Quick Sort – Rozděl a panuj

Slučovací metody

- Časová složitost: lineárně logaritmická
- Prostorová složitost (rekurze): logaritmická

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Quick Sort – Rozděl a panuj

Slučovací metody

- Časová složitost: lineárně logaritmická
- Prostorová složitost (rekurze): logaritmická
- Uspořádaná data – může být problematické

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

Merge Sort

- Slučování dvou proudů uspořádaných dat

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

Merge Sort

- Slučování dvou proudů uspořádaných dat
- Cyklické rozdělování a slučování

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

Merge Sort

- Slučování dvou proudů uspořádaných dat
- Cyklické rozdělování a slučování
- Lineárně logaritmická složitost

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

Merge Sort

- Slučování dvou proudů uspořádaných dat
- Cyklické rozdělování a slučování
- Lineárně logaritmická složitost
- Jeden krok slučování: práce se dvěma soubory, se dvěma frontami (seznamy)

Základní pojmy

Metody

Výběrové metody

Vkládací metody

Rozdělovací
metody

Slučovací metody

Merge Sort

- Slučování dvou proudů uspořádaných dat
- Cyklické rozdělování a slučování
- Lineárně logaritmická složitost
- Jeden krok slučování: práce se dvěma soubory, se dvěma frontami (seznamy)
- Sekvenční metoda, původně vytvořená pro přímou práci s magnetickými páskami