

Vyhledávání

Programovací techniky

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

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Vyhledávání: jedna z nejfrekventovanějších úloh

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Vyhledávání: jedna z nejfrekventovanějších úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Vyhledávání: jedna z nejfrekventovanějších úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Vyhledávání: jedna z nejfrekventovanějších úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:

- Vyhledávání: jedna z nejfrekventovanějších úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Vyhledávání: jedna z nejfrekventovanějších úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,
 - pozice dat.

- Vyhledávání: jedna z nejméně frekventovaných úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,
 - pozice dat.
- Důležitá je efektivnost **v dané aplikaci**

- Vyhledávání: jedna z nejméně frekventovaných úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,
 - pozice dat.
- Důležitá je efektivnost **v dané aplikaci**
- Datová složka + klíč (jednoduchý typ, řetězec)

- Vyhledávání: jedna z nejméně frekventovaných úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,
 - pozice dat.
- Důležitá je efektivnost **v dané aplikaci**
- Datová složka + klíč (jednoduchý typ, řetězec)
- V jednoduchém případě: data slouží i jako klíč

- Vyhledávání: jedna z nejméně frekventovaných úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,
 - pozice dat.
- Důležitá je efektivnost **v dané aplikaci**
- Datová složka + klíč (jednoduchý typ, řetězec)
- V jednoduchém případě: data slouží i jako klíč
- Ideální případ: Klíč je jednoznačný v rámci dané struktury

- Vyhledávání: jedna z nejméně frekventovaných úloh
- ATD Vyhledávací tabulka: jakákoliv struktura umožňující uložení dat a jejich efektivní vyhledání
- Typické operace: inicializace, vložení dat, vyhledání podle klíče
- Výsledek vyhledání:
 - logická hodnota,
 - pozice dat.
- Důležitá je efektivnost **v dané aplikaci**
- Datová složka + klíč (jednoduchý typ, řetězec)
- V jednoduchém případě: data slouží i jako klíč
- Ideální případ: Klíč je jednoznačný v rámci dané struktury
- Datové složky se stejným klíčem = synonyma

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup
- **Stromy**: stromový průchod = sekvenční přístup; uspořádané stromy: stromové hledání

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup
- **Stromy**: stromový průchod = sekvenční přístup; uspořádané stromy: stromové hledání
- **Pole**: sekvenční přístup; uspořádané pole: sekvenční přístup, půlení intervalu, indexace

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup
- **Stromy**: stromový průchod = sekvenční přístup; uspořádané stromy: stromové hledání
- **Pole**: sekvenční přístup; uspořádané pole: sekvenční přístup, půlení intervalu, indexace
- Dělení podle časové složitosti – metody:

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup
- **Stromy**: stromový průchod = sekvenční přístup; uspořádané stromy: stromové hledání
- **Pole**: sekvenční přístup; uspořádané pole: sekvenční přístup, půlení intervalu, indexace
- Dělení podle časové složitosti – metody:
 - lineární,

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup
- **Stromy**: stromový průchod = sekvenční přístup; uspořádané stromy: stromové hledání
- **Pole**: sekvenční přístup; uspořádané pole: sekvenční přístup, půlení intervalu, indexace
- Dělení podle časové složitosti – metody:
 - lineární,
 - logaritmické,

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Lineární struktury, nelineární struktury
- **Soubory** – velké množství dat: sekvenční přístup; uspořádané soubory: sekvenční přístup, půlení intervalu
- **Lineární seznamy** – optimalizace spotřeby paměti: sekvenční přístup; uspořádané seznamy: sekvenční přístup
- **Stromy**: stromový průchod = sekvenční přístup; uspořádané stromy: stromové hledání
- **Pole**: sekvenční přístup; uspořádané pole: sekvenční přístup, půlení intervalu, indexace
- Dělení podle časové složitosti – metody:
 - lineární,
 - logaritmické,
 - konstantní.

- Časová složitost lineární = nejhorší

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost lineární = nejhorší
- Všude dále v příkladech platí: pole P , s N složkami některého skalárního typu, rozměr pole může být $M \geq N$, hledaná hodnota C , logická proměnná **Nalezeno**.

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost lineární = nejhorší
- Všude dále v příkladech platí: pole P, s N složkami některého skalárního typu, rozměr pole může být $M \geq N$, hledaná hodnota C, logická proměnná Nalezeno.
- Sekvenční hledání obecně:

```
/*nastav se na začátek struktury*/  
while(/*není konec struktury*/ and /*nenalezeno*/)  
    /*jdi na další prvek*/;  
Nalezeno=/*není konec*/;
```

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost lineární = nejhorší
- Všude dále v příkladech platí: pole P, s N složkami některého skalárního typu, rozměr pole může být $M \geq N$, hledaná hodnota C, logická proměnná Nalezeno.

- Sekvenční hledání obecně:

```
/*nastav se na začátek struktury*/  
while(/*není konec struktury*/ and /*nenalezeno*/)  
    /*jdi na další prvek*/;  
Nalezeno=/*není konec*/;
```

- Sekvenční hledání konkrétně (v poli):

```
I=0;  
while (I<N and C!=P[I]) I++;  
Nalezeno=I<N;
```

Sekvenční hledání se zarážkou

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost stejná jako u předchozí metody

Sekvenční hledání se zarážkou

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost stejná jako u předchozí metody
- Na zarážku musí být volné místo (1 položka na konci)

Sekvenční hledání se zarážkou

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost stejná jako u předchozí metody
- Na zarážku musí být volné místo (1 položka na konci)
- Vložení zarážky je s konstantní složitostí

```
P[N]=C; //hledaná hodnota je zarážkou
I=0;    //dále sekvenční průchod
while (C!=P[I]) I++; //bude vždy nalezeno
Nalezeno=I<N; //otázkou pouze je, kde
```

Sekvenční hledání se zarážkou

ADT Vyhledávací
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se
zarážkou

Sekvenční hledání ve
vzestupně uspořádaném
poli

Logaritmické
metody

Indexace

Tabulka
s rozptýlenými
hesly

- Časová složitost stejná jako u předchozí metody
 - Na zarážku musí být volné místo (1 položka na konci)
 - Vložení zarážky je s konstantní složitostí
- ```
P[N]=C; //hledaná hodnota je zarážkou
I=0; //dále sekvenční průchod
while (C!=P[I]) I++; //bude vždy nalezeno
Nalezeno=I<N; //otázkou pouze je, kde
```
- Ušetření poloviny podmínky pouze zlepší implementační konstantu

# Sekvenční hledání v uspořádaném poli

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se  
zarážkou

Sekvenční hledání ve  
vzestupně uspořádaném  
poli

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Časová složitost opět lineární = nejhorší

# Sekvenční hledání v uspořádaném poli

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se  
zarádkou

Sekvenční hledání ve  
vzestupně uspořádaném  
poli

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Časová složitost opět lineární = nejhorší
- Velké plýtvání – nevyužívá se seřazené struktury

```
I=0;
while (I<N and C>P[I]) I++;
Nalezeno = I<N and C==P[I];
//možné díky zkrácenému vyhodnocení
```

# Sekvenční hledání v uspořádaném poli

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Sekvenční hledání

Sekvenční hledání se  
zarázkou

Sekvenční hledání ve  
vzestupně uspořádaném  
poli

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Časová složitost opět lineární = nejhorší
- Velké plýtvání – nevyužívá se seřazené struktury

```
I=0;
while (I<N and C>P[I]) I++;
Nalezeno = I<N and C==P[I];
//možné díky zkrácenému vyhodnocení
```

- Neúspěšné vyhledání je detekováno dříve, opět jen změna implementační konstanty, nikoliv charakteru algoritmu

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Rozmezí prvků pole se rozdělí na poloviny

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Rozmezí prvků pole se rozdělí na poloviny
- U středového prvku provedeme porovnání s hledanou hodnotou

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Rozmezí prvků pole se rozdělí na poloviny
- U středového prvku provedeme porovnání s hledanou hodnotou
- Na základě testu pak stejným způsobem pracujeme s levou, nebo s pravou polovinou daného rozmezí

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Rozmezí prvků pole se rozdělí na poloviny
- U středového prvku provedeme porovnání s hledanou hodnotou
- Na základě testu pak stejným způsobem pracujeme s levou, nebo s pravou polovinou daného rozmezí
- Konec neúspěšného hledání: rozmezí prvků je tvořeno jediným prvkem, který se nerovná hledané hodnotě

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Rozmezí prvků pole se rozdělí na poloviny
- U středového prvku provedeme porovnání s hledanou hodnotou
- Na základě testu pak stejným způsobem pracujeme s levou, nebo s pravou polovinou daného rozmezí
- Konec neúspěšného hledání: rozmezí prvků je tvořeno jediným prvkem, který se nerovná hledané hodnotě
- Časová složitost  $O(N) = k \cdot \log_2 N$

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Programové řešení:

```
int levy = 0; //interval hledání
int pravy = N-1;
int index = (levy+pravy)/2;
while (levy<pravy and p[index]!=C){
 if (C < p[index]) pravy=index-1; //jdeme vlevo
 else levy=index+1; //jdeme vpravo
 index=(levy+pravy)/2;
}
Nalezeno = p[index]==C;
```

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

**Stromové hledání:**

Indexace

Tabulka  
s rozptýlenými  
hesly

- Binární vyhledávací strom, uspořádání:  $ls \leq o \leq ps$

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

**Stromové hledání:**

Indexace

Tabulka  
s rozptýlenými  
hesly

- Binární vyhledávací strom, uspořádání:  $ls \leq o < ps$
- Implementace dynamickou strukturou:

```
pom=koren;
while (pom!=NULL and pom->klic!=C)
 if (pom->klic<C) pom=pom->ps;
 else pom=pom->ls;
Nalezeno = Pom!=NULL;
```

- Binární vyhledávací strom, uspořádání:  $ls \leq o < ps$
- Implementace dynamickou strukturou:

```
pom=koren;
while (pom!=NULL and pom->klic!=C)
 if (pom->klic<C) pom=pom->ps;
 else pom=pom->ls;
Nalezeno = Pom!=NULL;
```

- Schéma odpovídá sekvenčnímu hledání, probíhá ale napříč hladinami

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Půlení intervalu

Stromové hledání:

Indexace

Tabulka  
s rozptýlenými  
hesly

- Binární vyhledávací strom, uspořádání:  $ls \leq o < ps$
- Implementace dynamickou strukturou:

```
pom=koren;
while (pom!=NULL and pom->klic!=C)
 if (pom->klic<C) pom=pom->ps;
 else pom=pom->ls;
Nalezeno = Pom!=NULL;
```

- Schéma odpovídá sekvenčnímu hledání, probíhá ale napříč hladinami
- Pro vyvážený strom:  $O(N) = K \cdot \log_2 N$

- Binární vyhledávací strom, uspořádání:  $ls \leq o < ps$
- Implementace dynamickou strukturou:

```
pom=koren;
while (pom!=NULL and pom->klic!=C)
 if (pom->klic<C) pom=pom->ps;
 else pom=pom->ls;
Nalezeno = Pom!=NULL;
```

- Schéma odpovídá sekvenčnímu hledání, probíhá ale napříč hladinami
- Pro vyvážený strom:  $O(N) = K \cdot \log_2 N$
- Silná závislost na uspořádání vstupních dat – možná degradace na lineární seznam a kolaps na složitost  $O(N) = K \cdot N$

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání
- Silné omezení na datový typ hodnoty

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání
- Silné omezení na datový typ hodnoty
- Hodnoty nejsou složkami, ale *indexy*

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání
- Silné omezení na datový typ hodnoty
- Hodnoty nejsou složkami, ale *indexy*
- Přístupová funkce normuje hodnotu na indexový interval  $\langle 0; N - 1 \rangle$

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání
- Silné omezení na datový typ hodnoty
- Hodnoty nejsou složkami, ale *indexy*
- Přístupová funkce normuje hodnotu na indexový interval  $\langle 0; N - 1 \rangle$
- Implementace logickým polem – indikace přítomnosti

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání
- Silné omezení na datový typ hodnoty
- Hodnoty nejsou složkami, ale *indexy*
- Přístupová funkce normuje hodnotu na indexový interval  $\langle 0; N - 1 \rangle$
- Implementace logickým polem – indikace přítomnosti
- Implementace celočíselným polem – pro hodnoty, které se vícekrát opakují

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Nejrychlejší možná metoda vyhledání
- Silné omezení na datový typ hodnoty
- Hodnoty nejsou složkami, ale *indexy*
- Přístupová funkce normuje hodnotu na indexový interval  $\langle 0; N - 1 \rangle$
- Implementace logickým polem – indikace přítomnosti
- Implementace celočíselným polem – pro hodnoty, které se vícekrát opakují
- Hodnoty je potřebné do dané struktury uložit – lineární časová složitost; do samotného hledání se ovšem nezapočítává

- Logické pole, prohledávané hodnoty jsou v intervalu  $\langle X; Y \rangle$

```
const int X=-100;
const int Y=100;
typedef unsigned int Indexy;
Indexy Reindex(int I){ //přístupová funkce
 return I-X;
}
bool Pole[Y-X+1];
for (Indexy I=0; I<=Y-X; I++) Pole[I]=false;
int C; //naplnění
while (cin>>C) Pole[Reindex(C)]=true;
//hledání:
Nalezeno = Pole[Reindex(C)]
```

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data
- Rozptylování: funkce pracující v konstantním čase

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data
- Rozptylování: funkce pracující v konstantním čase
  - Celá a reálná čísla: (zaokrouhlení) + normalizace na interval pole

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data
- Rozptylování: funkce pracující v konstantním čase
  - Celá a reálná čísla: (zaokrouhlení) + normalizace na interval pole
  - Řetězce: manipulace s délkou a ordinálními hodnotami

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data
- Rozptylování: funkce pracující v konstantním čase
  - Celá a reálná čísla: (zaokrouhlení) + normalizace na interval pole
  - Řetězce: manipulace s délkou a ordinálními hodnotami
- Omezení počtu synonym: konstantní složitost, jinak lineární

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data
- Rozptylování: funkce pracující v konstantním čase
  - Celá a reálná čísla: (zaokrouhlení) + normalizace na interval pole
  - Řetězce: manipulace s délkou a ordinálními hodnotami
- Omezení počtu synonym: konstantní složitost, jinak lineární
- Implementace v existujících knihovnách – obecná rozptylovací funkce

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

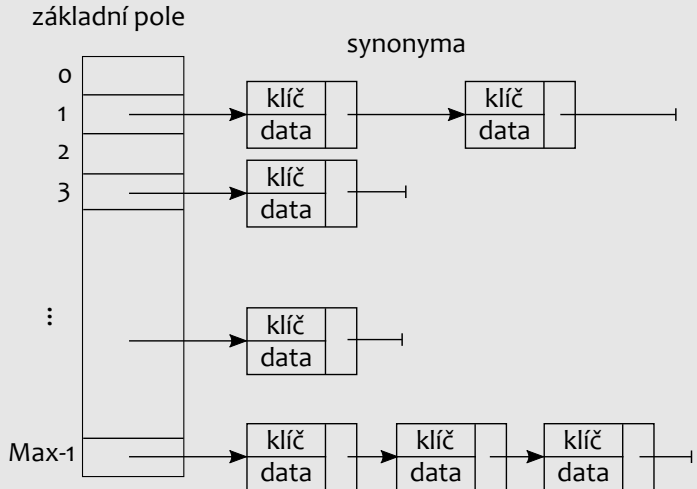
Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Kombinace indexace a sekvenčního hledání
- Jednoduchá vlastní implementace – optimalizace na konkrétní data
- Rozptylování: funkce pracující v konstantním čase
  - Celá a reálná čísla: (zaokrouhlení) + normalizace na interval pole
  - Řetězce: manipulace s délkou a ordinálními hodnotami
- Omezení počtu synonym: konstantní složitost, jinak lineární
- Implementace v existujících knihovnách – obecná rozptylovací funkce
- Asociativní pole `std::map` – poněkud odlišný pohled (implementováno typicky jako binární vyhledávací strom)

- Schéma:



ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

- Implementace:

```
const unsigned int Max=211;
typedef unsigned int Indexy;
typedef long int TypKlic;
struct Data {
 TypKlic Klic;
 void *DatSlozka;
};
struct Clen {
 Data D;
 Clen *N;
};
typedef Clen *UkClen;
typedef UkClen ZaklPole[Max];
typedef ZaklPole HashTable;
```

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

```
Indexy Hash(TypKlic K){
 return abs(K) % Max;
}
void Init(HashTable H){
 for (Indexy I=0; I<Max; I++) H[I]=NULL;
}
void HVloz(HashTable H, Data El){
 UkClen Pom;
 Indexy I = Hash(El.Klic);
 Pom = new Clen;
 Pom->D = El;
 Pom->N = H[I];
 H[I] = Pom;
}
```

ADT Vyhledávací  
tabulka

Možnosti realizace

Lineární metody

Logaritmické  
metody

Indexace

Tabulka  
s rozptýlenými  
hesly

```
UkClen HHledej(HashTable H, Data D){
 UkClen Pom;
 Indexy I = Hash(D.Klic); //indexace
 //následuje sekvenční hledání
 Pom = H[I];
 while (Pom!=NULL and Pom->D.Klic!=K)
 Pom = Pom->N;
 return Pom;
}
```