

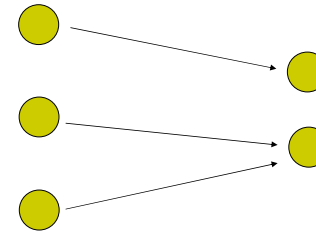
Vyhledávání

doc. Ing. Miroslav Beneš, Ph.D.
katedra informatiky FEI VŠB-TUO
A-1007 / 597 324 213
<http://www.cs.vsb.cz/benes>
Miroslav.Benes@vsb.cz



Pojem zobrazení (Map)

`Osoba vyhledejOsobu(String prijmeni) { ... }`



definiciční obor
(např. String)

obor hodnot
(např. Osoba)

Vyhledávání

2

Asociativní pole

- Zobecnění pojmu pole na indexy libovolného typu (index = vyhledávací klíč)
- Příklad (PHP):
 - `$dny = array();`
`$dny["Po"] = "Pondělí"; $dny["Út"] = "Úterý"; ...`
 - `echo $dny["St"];`

Vyhledávání

3

Rozhraní Map

```
interface Map {  
    // vložení hodnoty se zadaným klíčem  
    Object put(Object key, Object value);  
  
    // vyhledání podle klíče  
    Object get(Object key);  
  
    // odstranění hodnoty  
    Object remove(Object key);  
}
```

Vyhledávání

4

Implementace

- tabulka s lineárním vyhledáváním v poli nebo seznamu, $O(n)$
- tabulka s binárním vyhledáváním v poli, $O(\log n)$
- binární vyhledávací strom, $O(n) - O(\log n)$
- tabulka s rozptýlenými položkami. $O(1) - ??$

Vyhledávání

5

Implementace vyhledávací tabulky seznamem

```
class ArrayMap implements Map
{
    // klíče
    private ArrayList keys;
    // hodnoty
    private ArrayList values;

    public ArrayMap(int capacity) {
        keys = new ArrayList(capacity);
        values = new ArrayList(capacity);
    }
}
```

Vyhledávání

6

Implementace vyhledávací tabulky seznamem

```
public Object get(Object key)
{
    int index = keys.indexOf(key);
    if( index < 0 ) return null;

    return values.get(index);
}
```

Vyhledávání

7

Implementace vyhledávací tabulky seznamem

```
public Object put(Object key, Object value)
{
    int index = keys.indexOf(key);
    if( index < 0 ) {
        keys.add(key);
        values.add(value);
        return null;
    }

    Object oldValue = values.get(index);
    values.set(index, value);
    return oldValue;
}
```

Vyhledávání

8

Implementace vyhledávací tabulky seznamem

```
public Object remove(Object key)
{
    int index = keys.indexOf(key);
    if( index < 0 ) return null;

    Object oldValue = values.get(index);

    keys.remove(index);
    values.remove(index);
    return oldValue;
}
```

Vyhledávání

9

Tabulka s rozptýlenými položkami (hash table)

- Mapovací funkce
 - $h: \text{Object} \rightarrow 0 \dots (n-1)$
 - `Object get(Object key) { return tabulka[h(key)]; }`
 - Ideální mapovací funkce = prostá funkce
 - každý klíč se mapuje na jiný index
 - Skutečná mapovací funkce
 - mnohem více klíčů než indexů
 - vznikají kolize (synonyma)

Vyhledávání

10

Mapovací funkce v Javě

- class Object:
 - `int hashCode() { return h(key); }`
- Příklad:

```
class Osoba {
    String login;
    String jmeno;
    public int hashCode() {
        return login.hashCode();
    }
}
```

Vyhledávání

11

Řešení kolizí klíčů

- `index = key.hashCode() % kapacita`
- `indexkey1 = indexkey2 → kolize`
- Co s kolizemi?
 - Vyhnut se jim – pokud známe množinu klíčů předem (např. CD-ROM, klíčová slova jazyka, ...)
 - Minimalizovat počet kolizí a řešit je:
 - řetězení synonym do pomocných datových struktur (seznam, strom, ...)
 - otevřené adresování – nalezení náhradního indexu

Vyhledávání

12

Tabulka s řetězením synonym

```
class Entry {
    Object key;    // klíč
    Object value; // hodnota
    Entry next;   // další synonymum
}

Entry[] table
        = new Entry[kapacita];
```

Vyhledávání

13

Tabulka s řetězením synonym

```
public Object get(Object key)
{
    int index = key.hashCode() % table.length;
    HashMapEntry entry = table[index];

    while( entry != null ) {
        if( key.equals(entry.key) )
            return entry.value;
        entry = entry.next;
    }
    return null;
}
```

Vyhledávání

14

Tabulka s otevřeným adresováním

- Rozptylovací funkce
 - $r: 0..(kapacita-1) \rightarrow 0..(kapacita-1)$
- Příklady:
 - $r(i) = (i + 1) \% kapacita$
 - přechod na následující položku
 - $r(i) = (i + n) \% kapacita$
 - n musí být nesoudělné s kapacitou, např. prvočísla
 - je třeba využít celé kapacity tabulky

Vyhledávání

15

Tabulka s otevřeným adresováním

```
public Object get(Object key) {
    int index0 = key.hashCode()
                % table.length;
    int index = index0;
    Entry entry;
    while( (entry = table[index]) != null ) {
        if( key.equals(entry.key) )
            return entry.value;
        index = (index + delta) % table.length;
        if( index == index0 ) return null;
    }
    return null;
}
```

Vyhledávání

16

Vyhledávání v Javě

- `import java.util.*;`
- Rozhraní:
 - Map
 - Map.Entry
- Třídy:
 - HashMap
tabulka s rozptýlenými položkami
 - TreeMap
vyhledávací strom

Vyhledávání

17

Příklad

```
Map dny = new TreeMap();
dny.put("Po", "Pondělí");
dny.put("Út", "Úterý");
...

String s = (String)dny.get("St");

Iterator it = dny.entrySet().iterator();
// dny.keySet(), dny.values()
while( it.hasNext() ) {
    Map.Entry entry = (Map.Entry)it.next();
    System.out.println(entry.getKey());
}
```

Vyhledávání

18

Úkol do cvičení

- Vytvořte program, který pro zadaný soubor určí počty výskytů jednotlivých znaků a vytiskne je přehledně v tabulce.
 - a: 6
 - b: 2
 - c: 2
 - d: 1
 - ...

Vyhledávání

19