

**Igor Rožanc**

## **Osnove algoritmov in podatkovnih struktur I (OAPS I)**

**2. letnik, VSP Računalništvo in informatika, vse smeri**

**PROSOJNICE ZA 14. PREDAVANJA (4.1.2007)**

Študijsko leto 2006/07

### **Vrsta v Collection Framework (Java 1.5.0)**

**86**

**Queue<E>, PriorityQueue<E> in Comparator<E>:**

- Metode vmesnika **Queue<E>** in razreda **PriorityQueue<E>**:  
**element()**, **offer(E)**, **peek()**, **poll()**, **remove()**
- **PriorityQueue<E>** upošteva naravni red ali redefinirani **Comparator<E>**

```
PriorityQueue<Integer> pv1 = new PriorityQueue<Integer>(10);
PriorityQueue<Integer> pv2 = new PriorityQueue<Integer>(10,
    new Comparator<Integer>() {
        public int compare(Integer i, Integer j)
        { int rez = i%2-j%2;
          if (rez==0) rez=i-j;
          return rez;
        }
    } );
```

**Primer: razred `TestPriorityQueue.java`**

*Drevo* sestavlja množica elementov – *vozlišč*. Vozlišča so povezana na podlagi relacije, ki določa hierarhično strukturo.

**Primer:** kazalo v knjigi (relacija: vsebovanost) ...

**Formalna definicija:**

**Drevesna struktura z osnovnim tipom T je:**

- **prazna struktura** ali
- **vozlišče tipa T, kateremu je prirejeno končno število tujih drevesnih struktur z osnovnim tipom T (*poddreves*)**

**Ponazoritev dreves:** graf

**Primer grafa za odločitveno drevo:**

- pet zlatnikov, eden je ponaredek (je lažji)
- iščemo ga s tehtanjem

**Pojmi, ki so povezani z drevesi:**

- koren, vozlišče
- poddrevo, naddrevo
- naslednik, prednik
- neposredni naslednik (sin), neposredni predhodnik (oče)
- list (končni element), notranji element
- nivo, globina vozlišča
- višina drevesa
- stopnja vozlišča, stopnja drevesa (dvojiška, trojiška drevesa)
- pot do vozlišča (od korena)
- dolžina poti vozlišča, dolžina poti drevesa
- **Lastnosti (vrste) dreves:**
  - polnost
  - uravnoveženost
  - urejenost

## Lastnosti dreves - POLNOST:

- Za **izpolnjeno (complete) drevo** velja, da so stopnje vseh njegovih notranjih vozlišč enake (stopnji drevesa)
  - ugodno za prepis v tabelo, ker ni praznih delov tabele
- **Polno (full) drevo PD** je izpolnjeno drevo, ki ima vse liste na istem nivoju
  - PD (stopnja  $d$ , višina  $h$ ) ima število vozlišč  $n$ :  $(d^{h+1}-1) / (d-1)$
  - PD (stopnja  $d$ , število vozlišč  $n$ ) ima višino  $h$ :  $\log_a(n*d-n+1) - 1$

**Zgled:** polno dvojiško drevo ( $d=2$ ) in polno trojiško drevo ( $d=3$ ):

- |   |   |
|---|---|
| • $h=1: n = (2^{1+1}-1)/(2-1)=3$            | • $h=1: n = (3^{1+1}-1)/(3-1) = 4$          |
| • $h=2: n = 7$                              | • $h=2: n = 13$                             |
| • $h=10: n = 2047$                          | • $h=10: n = 88573$                         |
| • $n=1000: h = \log_2(1001)-1 = 9.97-1 = 9$ | • $n=1000: h = \log_3(2001)-1 = 6.92-1 = 6$ |
| • $10^6: h=19.93-1 = 19$                    | • $h=10^6: h=13.21-1 = 13$                  |