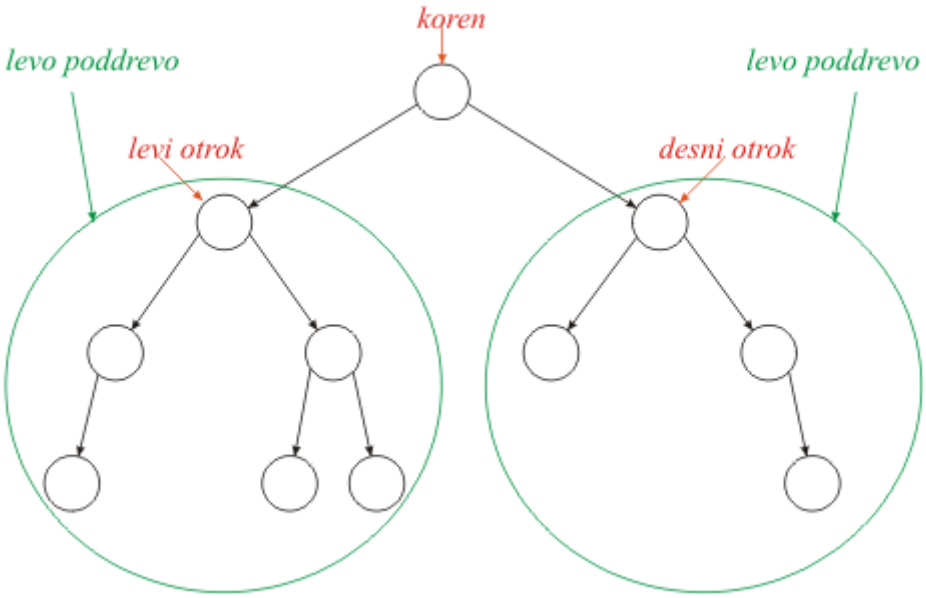
DVOJIŠKO DREVO

KAJ JE DVOJIŠKO DREVO?

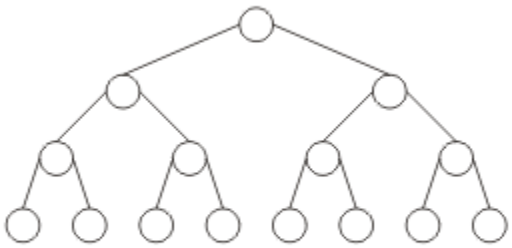
Dvojiško drevo je urejeno drevo, v katerem ima vsako vozlišče največ dva otroka (lahko ima tudi samo enega ali nobenega). Definirano je na končnem številu vozlišč, tako da velja ena od naslednjih trditev:

- V drevesu ni nobenega vozlišča (drevo je prazno).
- Drevo je sestavljeno iz treh disjunktnih množic: korena, levega poddrevesa in desnega poddrevesa.



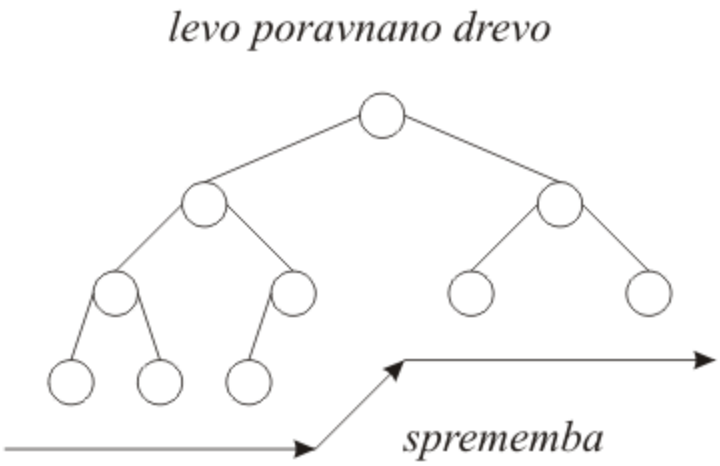
LASTNOSTI DVOJIŠKIH DREVES

Drevo globine h ima največ $\sum_{i=1}^{h} 2^{i-1} = 2^h - 1$ vozlišč.



nivo	št. otrok
nivo 1	≤ 1
nivo 2	≤ 2
nivo 3	≤ 4
nivo 4	≤ 8
nivo i	$\leq 2^{i-1}$

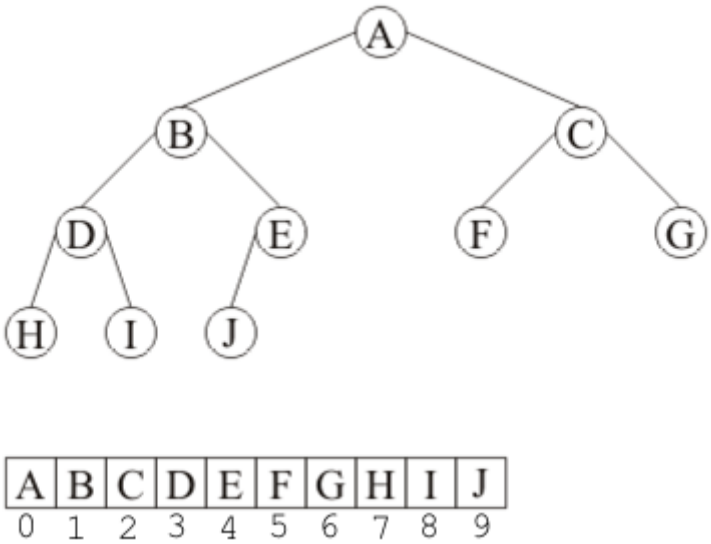
Enačaj velja, ko je drevo levo in desno poravnano (drevo je polno). Drevo je levo (desno) poravnano, če se nivoji listov razlikujejo za največ ena in če se ta sprememba, ko gremo iz leve (desne) proti desni (levi) zgodi največ enkrat. Polno dvojiško drevo je tisto, ki je hkrati levo in desno poravnano (zadnji nivo ima vsa možna vozlišča).



IMPLEMENTACIJA DVOJIŠKIH DREVES

STATIČNA PREDSTAVITEV DVOJIŠKIH DREVES

Za statično predstavitev dvojiških dreves lahko uporabimo polje z *n* elementi. To hkrati pomeni, da lahko v drevo shranimo največ *n* podatkov. Predstavitev vozlišč in ugotavljanje relacij starš, levi otrok, desni otrok je zelo preprosto. Označimo elemente polja od 0 naprej.



Poglejmo vozlišče na *i*-tem mestu:

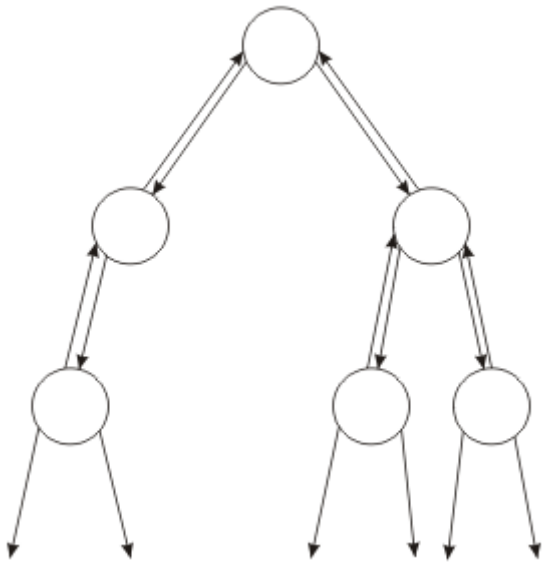
- levi otrok vozlišča i ima indeks *2*i+1*
- desni otrok vozlišča i ima indeks *2*i+2*
- starš vozlišča i ima indeks *(i-1)/2*, pri čemer gre za celoštevilsko deljenje

Omenjene formule veljajo za polja indeksirana od 0 naprej. Operacijo deljenja z 2 lahko zelo hitro izvedemo z desnim premikom bitov za eno mesto. Operacijo množenja z 2 pa lahko zelo hitro izvedemo z levim premikom bitov za eno mesto.

Statična predstavitev dreves je primerna za razmeroma polna drevesa (le tako je prostor dobro izkoriščen). Najmanj je primerno za izrojena drevesa (vsako vozlišče ima samo enega otroka). Običajno statično predstavljamo le levo oz. desno poravnana drevesa.

DINAMIČNA PREDSTAVITEV DVOJIŠKIH DREVES

Dvojiška drevesa si lahko predstavimo na dinamičen način tako, da zraven podatka v vozlišču shranimo tudi kazalec na levega in desnega otroka, dobro pa je imeti tudi kazalec na starša.



Deklaracija razreda za vozlišče v C++

```
class CVozlisce {
  public:
     tip vrednost; // tip je odvisen od vrste podatkov
     CVozlisce *levi;
     CVozlisce *desni;
     CVozlisce *stars;
};
```

PREGLED DVOJIŠKIH DREVES

Skozi vsa vozlišča drevesa se lahko sprehodimo na več načinov, ki določajo vrstni red (glede na vozlišče in njegova otroka) obdelovanja podatkov v vozlišču. Operacija *obdelaj(vozlisce)* je operacija, ki jo izvedemo nad podatkom vozlišča *vozlisce*. Poznamo tri standardne vrste pregleda dvojiških dreves:

- najprej starš (angl. pre-order) - S L D
- starš vmes (angl. in-order) - L S D
- starš potem (angl. post-order) - L D S

Te preglede najlažje implementiramo rekurzivno, lahko pa seveda tudi ne-rekurzivno. Poglejmo primer obeh pregledov.

REKURZIVNI PREGLED LSD

```
void SLD(CVozlisce* v) {
  if (v!=NULL) {
    SLD(v->levi);
    obdelaj(v);  // stori s podatkom, kar moraš
    SLD(v->desni);
  }
}
```

NEREKURZIVNI PREGLED LSD

```
void LSD(CVozlisce* v) {
  CSklad* sklad = new CSklad(); // pripravimo sklad za delo
  while (true) {
    while (v!=NULL) {
      sklad->vstavi(v); // na sklad damo vozlišče
      v = v->levi;
    }
    if (!sklad->prazen()) {
      v = sklad->odstrani();
      obdelaj(v);
      v = v->desni;
    } else break;
  }
  delete sklad;
}
```