Tema Nr. 6: Arbori Multicăi

Transformări între diferite reprezentări

Timp alocat: 2 ore

Implementare

Se cere implementarea **corectă** și **eficientă** a unor algoritmi de complexitate *liniară* pentru transformarea arborilor multicăi între următoarele reprezentări:

R1: *reprezentarea părinte*: pentru fiecare index, valoare din vector reprezintă indexul părintele, ex: $\Pi = \{2,7,5,2,7,7,-1,5,2\}$

R2: *reprezentare arbore multicăi*: fiecare nod conține cheia si un vector de noduri copil

R3: *reprezentare binara:* fiecare nod conține cheia si doi pointeri: unul către primul copil si al doilea către fratele din dreapta (ex: următorul frate).

Pentru *reprezentarea binara* (**R3**) trebuie sa implementați afișarea prietenoasa (**PP**).

Așadar, trebuie să definiți transformarea **T1** din reprezentarea *părinte* (**R1**) în reprezentarea *arbore multicăi* (**R2**), iar apoi transformarea **T2** în reprezentarea *binară* (**R3**). Folosiți afișarea prietenoasă pentru cele trei reprezentări (vezi pagina 2).

Definiți structurile de date. Puteți folosi structuri intermediare (ex: memorie adițională).

Notare și cerințe

- Implementarea corectă și pretty-print la **R1** – 5p
- Implementarea corectă la **T1** și pretty-print la **R2** – 1p
- Implementarea corectă și eficientă la **T1** în timp liniar – 1p
- Implementarea corectă la **T2** și pretty-print la **R3** – 2p
- Implementarea corectă și eficientă la **T2** în timp liniar – 1p

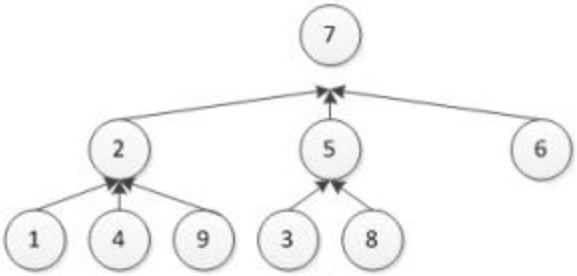
Evaluare

Corectitudinea algoritmilor va trebui demonstrată pe exemplul de la **R1** (Π). Folosiți afișarea prietenoasă pentru cele trei reprezentări.

Explicați ce structuri de date ați folosit pentru reprezentările **R2 și R3**.
Analizați eficienta în timp și spațiu a celor două transformări. Ați atins O(n) ? Ați folosit memorie adițională?

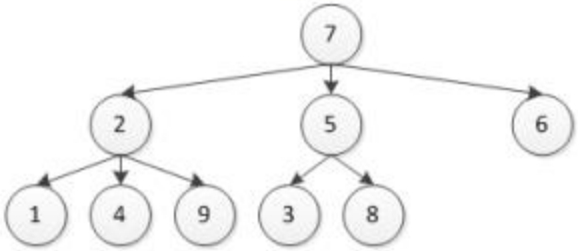
Input (R1): Π = {2, 7, 5, 2, 7, 7, -1, 5, 2}

1 2 3 4 5 6 7 8 9



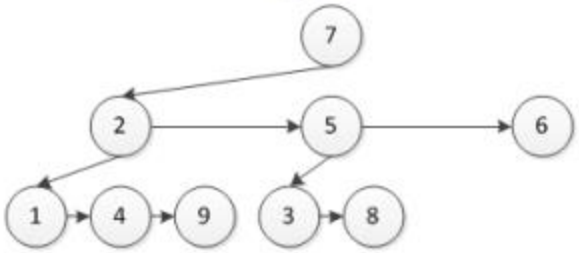
R1: parent representation

T1:parent -> multi-way



R2: multi-way representation

T2: multi-way -> binary



R3: binary representation

PP: prrety_print (binary)

```
7
    2
        1
        4
        9
5
        3
        8
    6
```

Pretty print