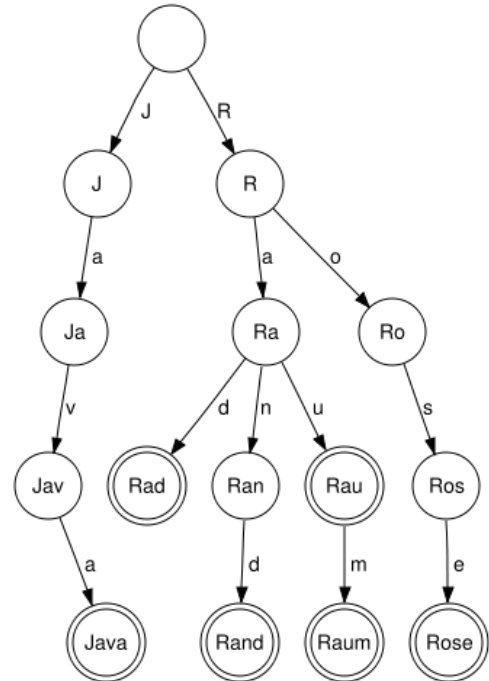


Imenik – Rješenje

Primjetimo prvo da sva slova jedne riječi u nazivu kontakta možemo zamijeniti njima odgovarajućim brojevima sa tastature. Time smanjujemo broj mogućih različitih znakova u riječi na samo 9. To nam mnogostruko smanjuje memorijske zahtjeve strukture koju ćemo koristiti za pretragu – *Trie*. *Trie* je stablo kod kojeg svaki čvor ima onoliko djece koliko je broj mogućih različitih znakova u stringovima koji se smještaju u njega. Pri tome svako dijete određuje po jedan znak. Put od korjena stabla do bilo kojeg čvora određuje po jednu riječ ili dio riječi. Slika ilustruje jedan takav primjer (na slici nisu prikazana djeca koja ne čine dio neke riječi). Pošto je moguće da jedna riječ bude prefiks druge riječi, u čvoru koji predstavlja kraj jedne riječi to i označimo.



Dakle, prvo unesemo sve riječi svih naziva kontakata u stablo. Nakon toga čitamo upite i vršimo prolazak kroz stablo krećući od korjena i prateći put koji označava string upita. Nakon što potrošimo sve znakove upita završit ćemo na nekom čvoru stabla – ili na praznom, tj. nepostojećem čvoru. Ako taj čvor postoji, onda trebamo ispisati u sortiranom poretku sve nazive koji imaju neku riječ koja se završava u podstablu kojem je korjen taj čvor na kojem smo završili.

Dalje možemo poboljšati vrijeme pretrage i zauzeće memorije tako što umetanje riječi u stablo obustavimo nakon što unesemo prvih 10 znakova te riječi, sa obzirom da upiti neće imati više od 10 znakova, pa nam nije bitno da gradimo strukturu stabla na dubljim nivoima.

Pretraga po upitu se vrši u $O(L)$ vremenu, gdje je L dužina upita.