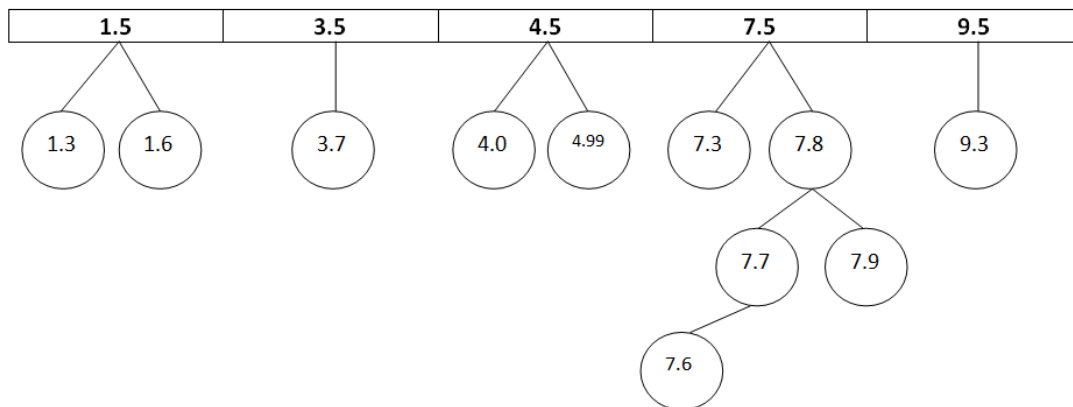


Rozważmy strukturę danych, której korzeń jest tablicą o długości  $N$ , której każdy element jest korzeniem poddrzewa. Poddzewa tworzone są zgodnie z definicją drzew wyszukiwań binarnych. Załóżmy, że każdy element tablicy (korzenia) przechowuje liczby od 0.5 do  $+\infty$  z krokiem 1.0, natomiast poddrzewa liczby ułamkowe (z dokładnością do 0.01), które są oddalone na osi liczbowej nie dalej niż 0.5 od wartości korzenia.



Rysunek 1: Przykład rozważanej struktury danych

Przedstawiona struktura w efektywny sposób umożliwia przechowywanie rozłącznych, nieposortowanych zbiorów danych. Czy znasz jakieś przykłady wykorzystania w których mogłaby znaleźć zastosowanie?

**Zadanie 1** (0.75 pkt) Zaimplementuj przedstawioną we wstępie strukturę danych oraz narzędzie do tekstowej wizualizacji danych w sposób jaki pokazano poniżej:

```

1.5-1.3
  -1.6
3.5-3.7
4.5-4.0
  -4.99
7.5-7.3
  -7.8--7.7---7.6
    --7.9
9.5-9.3
  
```

Elementy znajdujące się na tym samym poziomie powinny znajdować się w tej samej kolumnie. Kolejne poziomy rozgraniczone są liczbą znaków rozdzielających '-' odpowiadającą poziomowi zagnieźdżenia.

W celu zaprezentowania wizualizacji wypełnij strukturę za pomocą przykładowych danych.

**Zadanie 2** (0.75 pkt) Proszę założyć początkowy stan struktury zgodnie z przykładem z rys. 1.

Zaimplementuj następujące operacje:

*INSERT*( $x$ ) - wstawianie węzła o wartości  $x$  zgodnie z założeniami struktury.

*MINIMUM*( $y$ ) - wyszukiwanie najmniejszej wartości węzła w poddrzewie reprezentowanym przez korzeń  $y$ . Dla ułatwienia można posłużyć się liczbami całkowitymi (np. 4 zamiast 4.5) przy wyborze korzenia.

*MAXIMUM*( $y$ ) - wyszukiwanie największej wartości węzła w poddrzewie reprezentowanym przez korzeń  $y$ . Dla ułatwienia można posłużyć się liczbami całkowitymi (np. 4 zamiast 4.5) przy wyborze korzenia.

*SEARCH*( $x$ ) - sprawdzenie czy węzeł o wartości  $x$  jest zapisany w strukturze.

**Zadanie 3** (0.5 pkt) Sprawdź czas wykonania wszystkich operacji z zadania 2 dla struktury zawierającej np. 25, 50, 100, 500 i 1000 elementów (losowe dane). Oszacuj ich złożoność obliczeniową

---