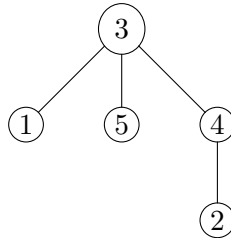


## Zadanie 1 Ciąg Prüfera

Drzewo  $T$ , w którym wierzchołki oznaczono kolejnymi liczbami naturalnymi od 1 do  $n$ , zdefiniowano za pomocą zbioru wierzchołków  $V$  i zbioru krawędzi  $E \subset \{\{a, b\} : a, b \in \{1, 2, \dots, n\}\}$ . Ciąg Prüfera jest ciągiem  $n - 2$  liczb wybranych ze zbioru  $\{1, 2, \dots, n\}$ , który określa dokładnie jedno drzewo oznaczone mające  $n$  wierzchołków. Ciąg ten uzyskujemy na podstawie drzewa  $T$  iteracyjnie usuwając jego wierzchołki, aż zostaną tylko dwa. Mówiąc bardziej precyzyjnie, w  $i$ -tym kroku algorytmu generowania ciągu Prüfera usuwany jest liść oznaczony najmniejszą liczbą, a  $i$ -tym elementem ciągu Prüfera staje się liczba oznaczająca jego sąsiada. Ciąg Prüfera drzewa dwuwierzchołkowego ma długość zero. Na poniższym rysunku przedstawiono drzewo odpowiadające ciągowi 343.



W języku C++ z wykorzystaniem odpowiednich kontenerów STL zaimplementuj funkcję

```
vector<int> Pru(set<int> &V, set<pair<int, int> > &E)
```

zwracającą ciąg (tablicę) Prüfera dla danego drzewa  $T = (V, E)$ .

## Zadanie 2 Konstruowanie drzewa oznaczonego

Dany jest zbiór wierzchołków  $V = \{1, 2, \dots, n + 2\}$  oraz początkowo pusty zbiór krawędzi  $E$ . Poniższy algorytm określa zbiór krawędzi (konstruuje drzewo) na podstawie ciągu Prüfera.

INVPRU( $(b_1, b_2, \dots, b_n), V$ )

- 1 **if** ciąg  $(b_1, \dots, b_n)$  jest pusty ( $|V| = 2$ )
- 2     **then** dodaj do  $E$  krawędź  $\{x, y\}$ , gdzie  $V = \{x, y\}$
- 3     **else**  $m \leftarrow \text{MIN}(V - \{b_1, b_2, \dots, b_n\})$
- 4         Dodaj do  $E$  krawędź  $\{m, b_1\}$
- 5         INVPRU( $(b_2, \dots, b_n), V - \{m\}$ )



Zakładając, że zbiór krawędzi  $\text{set}<\text{pair}<\text{int}, \text{int}> > E$  zadeklarowany jest globalnie, zaimplementuj procedurę INVPRU w języku C++ z wykorzystaniem odpowiednich kontenerów STL.