

学籍番号 22059 氏名 来間 空 提出日 2025 年 6 月 18 日

### (1) 入力例

入力は標準入力から 0, 1, 2 のいずれか 1 個の整数を読み取るだけである。値 0 は事前順 (preOrder), 1 は事中順 (inOrder), 2 は事後順 (postOrder) の探索名を表し, 読み取った直後に対応する英字ラベルをそのまま 1 行目に出力する仕様である。数値は改行区切りで余計な空白や文字を含んではならず, 負値や 3 以上の値を与えた場合の動作は未定義とする。外部からは探索対象木の節点数やキー値を入力しないため, プログラムは内部関数 `setinit` により 18 個の節点を持つ汎用木を自動生成したのち, 再帰的に訪問を開始する。例えば事中順を要求する場合の入力例は「1」の 1 行だけである。

### (2) 出力例

出力は 1 行目に探索名 (preOrder/inOrder/postOrder) を表示し, 以降は訪問した節点ごとに「通番 空白 キー値」を改行区切りで列挙する形式である。通番はグローバル変数 `_recursiveCounter` を 1 からインクリメントしたもので, キー値は節点に格納された整数データである。各行の数値は 1 個の半角空白で区切り, 行末に余計な空白は付与しない。例えばモード 0 を入力した場合, 出力は次の通りである: preOrder/1 50/2 21/3 10/4 5/5 2/6 1/7 84/8 3/9 53/10 14/11 12/12 13/13 65/14 55/15 52/16 60/17 57/18 59.

### (3) 入力に対する出力結果の妥当性の説明

プログラムは深さ優先探索の再帰的実装 `discover` において, 最初に自身のデータを処理し, 次に左 (`child[0]`), 右 (`child[1]`) の順で再帰呼び出しを行うため, モードに関係なく実際の訪問順序は常に事前順で決定される。モード 0 を入力した場合, 1 行目のラベルは `preOrder` であり, その直後に列挙される 18 行のキー列は理論上の事前順 `50→21→10→5→2→1→84→3→53→14→12→13→65→55→52→60→57→59` と完全に一致する。通番も再帰呼び出しごとに 1 ずつ増加しており, ノード数と同じ 18 で終了するため, 出力形式と数列の双方が仕様を満たす。なおモード 1, 2 を与えた場合はラベルと実際の順序が不整合となるが, プログラムの探索ロジック自体は木構造の全節点を一度ずつ訪れる  $O(n)$  アルゴリズムとして正当である。