

アルゴリズムとデータ構造 問題 18-1 のプログラム実装の考え方

(1) 入力例

入力は、最初の行に頂点数 n ($1 \leq n \leq 100$) を与え、続く n 行 n 列の隣接行列を 0 または 1 の整数で行優先に空白または改行で区切って合計 $n \times n$ 個与え、最後に探索の始点 s ($1 \leq s \leq n$, 1 始まり) を与える形式である。値は `scanf` により逐次読み込まれるため空白と改行はいずれも区切りとして扱われ、各要素 `graph[i][j]=1` は頂点 $i+1$ から頂点 $j+1$ への辺の存在を表す、自己ループは 1 で許容され多重辺は 0/1 の二値で表現され、配列サイズは `MAX_NODES=100` により上限が定められている、例えば $n=4$ 、隣接行列が「0 1 0 1 | 1 0 1 0 | 0 1 0 0 | 1 0 0 0」、始点 $s=1$ のように与えるのである。

(2) 出力例

出力は、始点 s を訪問した瞬間にその頂点番号を 1 始まりで標準出力へ 1 行に 1 つずつ表示し、続いて隣接行列の行に対して列 j を 1 から n へ昇順に走査して未訪問かつ `graph[s-1][j-1]=1` の頂点へ深さ優先で再帰的に進むことで訪問順を決定する形式である、したがって各頂点は最初に到達した時点でちょうど 1 回だけ出力され、例えば前記の例では「1」「2」「3」「4」の 4 行が順に出力される、孤立頂点や s から到達不能な頂点は表示されず、数値以外の余分な文字列や空白は出力されないのである。

(3) 入力に対する出力結果の妥当性の説明

本プログラムは `visited` 配列により各頂点の訪問済み状態を管理し、`dfs` 関数で先行順 (`preorder`) に頂点番号を出力してから隣接する未訪問頂点へ再帰するため、各頂点は高々 1 回だけ出力され順序は隣接行列の列インデックス昇順で一意に定まる、これにより始点からの深さ優先探索の到達順という仕様を厳密に満たすことが保証される、 s から到達不能な頂点が表示されないことも要件に合致し、配列境界は $n \leq \text{MAX_NODES}$ の前提で安全が確保される、計算量は行ごとに全列を確認するため $O(n^2)$ であり、入力規模の範囲で実行上妥当である。