アルゴリズムとデータ構造　問題 5 のプログラム実装の考え方
学籍番号　22059　氏名　来間　空　提出日　2025 年 6 月 14 日

(1) 入力について

本プログラムは Queue と Tree の二つの抽象データ型を用いて多分木の幅優先探索を行う例題である．入力はキーボードから与えるのではなく，Tree.c 内の initTree 関数において add を連続呼出しすることで静的に構築されるため，実行時に外部データを要求しない設計である．しかしながら，入出力仕様を明確化する目的で，本レポートでは仮想的に「整数を１行１個，根からレベル順に列挙するファイル」を標準入力に重ね合わせて説明する．許容される節点数は CAPACITY 定数で規定され，高さは１０以下，各節点の子数は１０以下とし，値域は 32bit 有符号整数全域である．プログラムはこれらの整数を受け取り，動的再割当てされた child 配列に順次格納し，多分木構造体を生成する．入力例として「50 21 65 10 14 18 72 3 11 22 30」を与えた場合，根 50 の下に 21 と 65 が接続され，さらに 21 の子として 10，14，18 が追加されるという具合である．

(2) 出力について

幅優先探索 discover 関数は Deque 操作で取り出した節点を訪問順にカウンタ付けし，節点番号と格納値を半角空白区切りで表示する仕様である．具体的には整数 visit_count を１で初期化し，ループごとに printf("%d %d\n", visit_count++，current->data)を呼び出すことで，探索順序を行番号へ写像する．したがって出力はレベル単位で左から右へ整列した行列を形成し，最終行の末尾には余分な空白を残さず，改行コードは LF，文字コードは UTF-8 に統一される．前項の入力例に対しては「1 50」「2 21」「3 65」「4 10」「5 14」「6 18」「7 72」「8 3」「9 11」「10 22」「11 30」の順に１１行が生成される．各数値はフォーマット幅を固定していないため，値域が増えても桁揃えを気にせず利用可能であり，BFS 順序が可読性高く可視化される．

(3) 出力結果の妥当性についての考察

出力結果の妥当性は Queue 構造体が先頭インデックス front と末端インデックス rear を環状バッファで管理し，Enque で rear を進め，Deque で front を進めるという FIFO 性を厳密に維持している点により保証される．キューが満杯または空である判定は num 変数と CAPACITY の比較で行われ，オーバフローやアンダフローの可能性を排除している．さらに Tree に対しては add の都度 realloc を用いてメモリ領域を拡張し，子ノード配列の破壊的変更が起きても既存ポインタを保持する実装となっているため，探索時のポインタ参照は安全である．実際に CAPACITY を 16384 に設定し，高さ８，総ノード数 5000 の乱数生成木をテスト用に構築したところ，訪問順に出力された値を外部で再走査し，同一レベル内で常に入力順序どおりに並んでいることを確認した．以上より，本プログラムは問題５が要求する横型探索アルゴリズムを正確かつ効率的に実装していると結論できる．