

問題 21-1 プログラムの考え方

学籍番号 22059 氏名 来間 空 提出日 2025 年 12 月 10 日

アルゴリズムとデータ構造 問題 21-1 のプログラム実装の考え方

(1) 入力例

本プログラムでは、ユーザからのキーボード入力は一切用いず、 8×8 のチェス盤と 8 個のクイーンを前提条件として内部で固定的に扱う仕様である。main 関数ではまず盤面配列 `board[8][8]` を `init_board` 関数で 0 に初期化し、クイーンの列位置を格納する `queen_position[8]` 配列を用意した上で、再帰関数 `set_queen` を行 0 から呼び出して全ての配置パターンを探索する。したがって、実行時に高さや個数を指定したり数値を入力する場面は存在せず、入力例も「入力なし」となる。入力仕様としては、8 行 8 列の盤面サイズとクイーン 8 個という問題設定自体が暗黙の入力であり、プログラム中では行と列を 0~7 の整数インデックスで表現することで探索を行っている。

(2) 出力例

出力は、クイーンの配置が攻撃し合わない全ての解について、8 行 8 列の盤面を 0 と 1 で表した表形式で順に表示する仕様である。`print_board` 関数では、行ごとに列 0~7 を走査し、その行にクイーンが置かれている列インデックスと一致する位置だけ 1 を、それ以外を 0 として出力し、各数値の間を 1 文字の空白で区切った後に改行する。例えば、1 行目の 3 列目にクイーンがある場合、1 行目は「0 0 1 0 0 0 0 0」のように表示される。8 行分を出力した後に、現在までに見つかった解の番号である `count` の値を 1 行で表示し、区切りとして「-----」を続けて印字する。この形式が全ての解に対して繰り返され、最終的には 8 クイーン問題に既知の全 92 通りの解が番号付きで列挙される出力仕様となっている。

(3) 入力に対する出力結果の妥当性の説明

本プログラムでは、各行に 1 個ずつクイーンを置くという制約の下で、縦横斜めに互いを攻撃しない配置のみを列挙するバックトラッキング探索を行っている。`set_queen` 関数は、引数 `row` で現在処理中の行を表し、各列 `col` について `board[row][col]` が 0 すなわち他のクイーンから攻撃されていないマスだけを候補として選び、`queen_position[row]` に列番号を記録する。続いて `change_board` 関数でそのマスの縦、横、斜め方向の全マスに +1 を加えて使用不能とし、次の行 `row+1` に対して再帰呼び出しを行う。再帰から戻ったら同じマスに対して -1 を加えて元の状態に戻すことで、他の候補列の検査に影響が残らないようにしている。`row` が 8 に達したときには 8 行全てにクイーンが 1 つずつ安全に配置されているため、このときだけ `count` をインクリメントして `print_board` で出力する。この処理によって、攻撃可能な配置は途中で枝刈りされ、

問題 21-1 プログラムの考え方

学籍番号 22059 氏名 来間 空 提出日 2025 年 12 月 10 日

条件を満たす全ての解だけが重複なく番号付きで表示されるため、問題 21-1 で求められている 8 クイーン問題の全解列挙という要件を正しく満たしているといえる。