**Báo cáo BTL 1: Trò chơi Shikaku**

**1. Giới Thiệu**

Shikaku là một bài toán chia bảng lưới thành các hình chữ nhật, trong đó mỗi hình chữ nhật chứa chính xác một số cho trước và diện tích hình bằng số đó.

**2. Dữ Liệu Đầu Vào**

**Cấu trúc dữ liệu:**

* Dòng đầu: Số hàng rows, số cột cols.
* Dòng tiếp theo: Danh sách các số nguyên dương trong lưới.
* Các dòng tiếp theo: Lưới [rows x cols], trong đó:
  + -: Ô trống.
  + Số nguyên dương: Yêu cầu mỗi vùng hình chữ nhật cần bao phủ diện tích bằng số đó.

**Ví dụ đầu vào:**

11

13

- - 6 - - - - - - - 10 - -

- - - - 4 - - - 2 - - - -

- 9 - - - - 6 - - - - 8 -

- - - 3 - - - - - 3 - - -

- - - - - 4 - 3 - - - - -

6 - 3 - - - - - - - 9 - 6

- - - - - 4 - 2 - - - - -

- - - 3 - - - - - 2 - - -

- 4 - - - - 12 - - - - 10 -

- - - - 6 - - - 6 - - - -

- - 8 - - - - - - - 4 - -

**3. Quá Trình Giải Thuật**

**Bước 1: Tiền xử lý dữ liệu**

* Đọc dữ liệu từ file.
* Xác định các vị trí chứa số nguyên dương.
* Khởi tạo các danh sách hỗ trợ như factors\_cache (lưu ước số chẵn cách chia diện tích cho từng vùng).

**Bước 2: Xác định các vùng**

* Duyệt qua từng số nguyên dương.
* Xác định cách chia hợp lệ dựa trên các ước số.
* Kiểm tra tính hợp lệ của mỗi vùng (đảm bảo không chồng lấn).

**Bước 3: Tìm kiếm bằng DFS**

**Giải thuật DFS là gì?**

DFS (Depth-First Search - Tìm kiếm theo chiều sâu) là một thuật toán duyệt hoặc tìm kiếm trên đồ thị hoặc cây. DFS hoạt động bằng cách bắt đầu từ một nút gốc, sau đó tiếp tục mở rộng theo chiều sâu trước khi quay lại để duyệt các nhánh khác.

Trong bài toán Shikaku, DFS được sử dụng để thử các cách chia vùng và quay lui khi gặp trường hợp không hợp lệ. Cách thức hoạt động cụ thể như sau:

1. **Bắt đầu từ vị trí chứa số nguyên dương đầu tiên** trong lưới.
2. **Duyệt qua tất cả các cách chia hình chữ nhật hợp lệ** dựa trên các ước số của số đó.
3. **Kiểm tra xem vùng đó có thể đặt vào lưới hay không**, tức là không bị chồng lấn với các vùng đã đặt trước đó.
4. **Nếu hợp lệ**, đánh dấu vùng đó vào lưới và tiếp tục tìm kiếm sâu hơn bằng cách gọi đệ quy DFS cho số tiếp theo.
5. **Nếu không tìm được cách chia hợp lệ**, thuật toán sẽ quay lui (backtrack) để thử cách chia khác.
6. **Quá trình tiếp tục cho đến khi tất cả số nguyên dương đều được phân vùng hợp lệ hoặc không có cách nào khả thi.**

DFS giúp đảm bảo rằng tất cả các cách chia có thể đều được thử, và khi gặp bế tắc, thuật toán sẽ quay lui để thử hướng khác.

**4. Kết Quả Đầu Ra**

**Mảng kết quả:**

Mỗi vùng được đánh số duy nhất.

**Ví dụ đầu ra:**

0 0 0 1 1 1 1 1 1 1 1 1 1

0 0 0 7 2 2 2 2 3 6 6 6 6

4 4 4 7 5 5 5 10 3 6 6 6 6

4 4 4 7 5 5 5 10 8 8 8 14 14

4 4 4 9 9 9 9 10 13 13 13 14 14

11 11 12 17 22 15 15 16 13 13 13 14 14

11 11 12 17 22 15 15 16 13 13 13 21 21

11 11 12 17 22 20 20 20 18 18 25 21 21

19 19 19 19 22 20 20 20 23 23 25 21 21

24 24 24 24 22 20 20 20 23 23 25 21 21

24 24 24 24 22 20 20 20 23 23 25 21 21

**5. Kết Luận**

* Giải thuật được triển khai bằng DFS, kèm theo bộ nhớ cache cho các ước số.
* DFS giúp tìm kiếm toàn bộ không gian các phương án có thể, đảm bảo tìm ra cách chia hợp lệ.
* Hiệu suất giải thuật tùy thuộc vào kích thước lưới và số lượng số nguyên dương cần xử lý.