

Mục lục

Bài A. VILLAGE	2
Bài B. PYRAMID	3
Bài C. HASHIRA	4
Bài D. TRIANGLE	5

Bài A. VILLAGE

Decomposition:

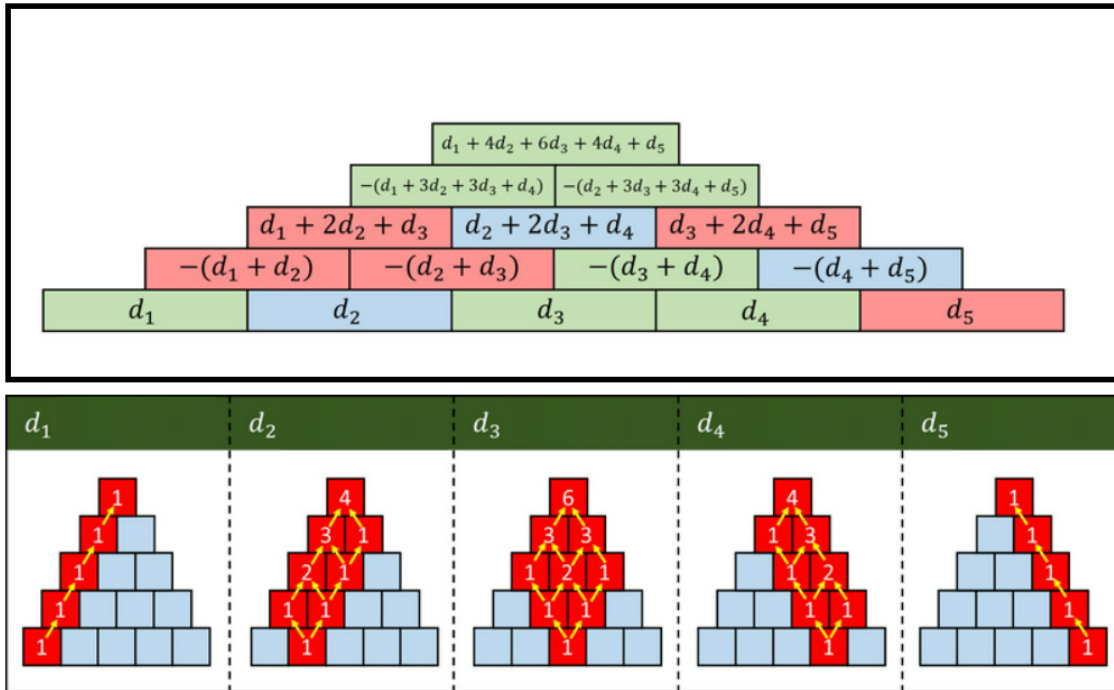
- Bài này sẽ sử dụng kĩ thuật bitmask.
- Tuy nhiên, quy hoạch động (QHD) thì vẫn là QHD.
- Ta cần phải xác định trạng thái bao gồm những gì, các trạng thái sẽ đi theo thứ tự nào (trạng thái nào trước, trạng thái nào sau), đâu là trạng thái đầu, đâu là trạng thái cuối.
- **Một trạng thái sẽ bao gồm:** các đỉnh đã đi qua, đỉnh hiện tại đang ở.
- **Trạng thái đầu:** các đỉnh đi qua chỉ có 1, đỉnh hiện tại đang ở là 1.
- **Trạng thái cuối:** đã đi qua tất cả các đỉnh, đỉnh hiện tại đang ở là n .
- **Số trạng thái:** $\sum_{i=1}^n 2^i \times i \approx n \times 2^n$.
- **Chi phí chuyển trạng thái:** $O(n)$ (tại sao ?)
- **Độ phức tạp:** $O(n^2 2^n)$

Cụ thể:

- Biểu diễn các tập con của tập hợp các số từ 1 đến n thành dãy bit độ dài n , lưu dưới dạng số nguyên cho thuận tiện lưu trữ.
- Ví dụ $n = 4$ thì $mask = 3_{10} = 0011_2$ sẽ mang ý nghĩa $mask$ chứa đỉnh 1 và 2.
- Gọi $dp_{mask,i}$ là số cách di chuyển từ thành phố 1, đi qua các thành phố có trong $mask$ và thành phố cuối của đường đi là thành phố i . Tất nhiên, i phải có trong $mask$.
- $dp_{mask,i} = \sum_{j \in mask} dp_{mask-2^i,j} + c_{i,j}$. Với $c_{i,j}$ là số đường đi trực tiếp từ i đến j .
- Kết quả: $dp_{2^n-1,n}$.
- Có thể optimize bằng cách chỉ tính dp cho $n-1$ thành phố đầu và lấy $ans = \sum_{i=2}^{n-1} dp_{2^{n-1}-1,i} \times c_{i,n}$.

Bài B. PYRAMID

Nếu coi 3 màu lần lượt là các số 0, 1, 2 thì một viên gạch được đặt lên 2 viên gạch có số là x, y bất kì sẽ có số là $[-(x + y)] \pmod 3$. Thử nhận xét trên với một trường hợp nhỏ (khoảng 5) ta được kết quả như sau:



Đây là dạng rất quen thuộc của Nhị thức Newton, trong đó số thứ i sẽ được thêm vào ô hàng trên cùng $C_{n-1}^{(i-1)}$ lần. Và để ý dấu của biểu thức, ta sẽ suy ra được công thức của ô ở hàng trên cùng:

$$(-1)^{n-1} \left(\sum_{i=1}^n s_i \times C_{n-1}^{i-1} \right) \bmod 3$$

Việc cuối cùng cần làm là tính C_{n-1}^i nhanh mà thôi! Có thể tính theo cách truyền thống, tuy nhiên có thể tính toán đơn giản hơn, chủ yếu là dựa vào tính chất sau:

$$\frac{1}{x} \equiv x \pmod{3}$$

$$\begin{aligned} C_{n-1}^i &= \frac{(n-1)!}{i! \times ((n-1) - i)!} \\ &= \frac{(n-1)! \times ((n-1) - (i-1))}{(i-1)! \times i \times ((n-1) - (i-1)!) } \\ &= \frac{(n-i)}{i} \times C_{n-1}^{i-1} \end{aligned}$$

Độ phức tạp $O(n)$.

Bài C. HASHIRA

Bài này tư tưởng rất đơn giản:

- Tìm giao điểm của hai đường thẳng tạo thành vùng chiếu của camera với đa giác.
- Tính diện tích đa giác vùng chiếu (gồm camera, giao điểm bên trái, đi theo chiều kim đồng hồ cho đến khi gặp giao điểm bên phải), tính diện tích đa giác ban đầu.
- Chia lấy tỉ lệ.

Tuy nhiên trong quá trình decomposition sẽ gặp các bài toán nhỏ sau:

- Tính diện tích đa giác.
- Tính giao điểm hai đường thẳng.
- Kiểm tra một điểm có thuộc đoạn thẳng hay không.
- Xoay đường thẳng quanh một điểm.
- Tính khoảng cách giữa hai điểm.

Bài D. TRIANGLE

Đây là bài exxtra, submit tại VNOI, mã bài là MTRIAREA.

Đầu tiên, có một nhận xét quan trọng cho bài này, đó là các đỉnh của tam giác kết quả phải thuộc bao lồi của tập điểm. Các bạn có thể cố gắng tự chứng minh (Tuấn Anh cũng không biết chứng minh).

Sau đó, chọn hai điểm trên bao lồi, tương ứng với hai điểm đó, ta sẽ tìm điểm còn lại (nằm giữa hai điểm đó trên bao lồi) sao cho 3 điểm tạo thành tam giác có diện tích lớn nhất. Có thể chặt nhị phân hoặc hai con trỏ để tìm điểm thứ 3 đó.