

# ĐỀ LUYỆN TẬP NÂNG CAO – KỸ NĂNG XỬ LÝ CÁC BÀI TẬP NHIỀU SUBTASK

Thứ 3, ngày 13 tháng 08 năm 2024.

## BÀI 1. CẮT GIẤY

Alice có một mảnh giấy hình chữ nhật có cạnh dọc và ngang tương ứng là  $a$  và  $b$  đơn vị. Mảnh giấy được chia thành  $a \times b$  ô vuông đơn vị, chiều dọc chia thành  $a$  hàng, các hàng được đánh số từ 1 đến  $a$  từ trên xuống dưới, chiều ngang được chia thành  $b$  cột từ 1 đến  $b$  từ trái sang phải. Ô nằm giao giữa hàng  $i$  ( $1 \leq i \leq a$ ) và cột  $j$  ( $1 \leq j \leq b$ ) được gọi là ô  $(i, j)$ . Có  $c$  ô vuông của mảnh giấy đã được tô màu. Với cặp số  $(d, e)$ , Alice muốn chọn một hình vuông trên mảnh giấy ban đầu và cắt mảnh giấy hình vuông này từ mảnh giấy ban đầu thỏa mãn điều kiện sau:

- Mảnh giấy chứa nguyên các ô vuông của mảnh giấy ban đầu;
- Mảnh giấy có cạnh chia hết cho  $d$ ;
- Mảnh giấy có không quá  $e$  ô đã tô màu.

**Yêu cầu:** Cho thông tin về mảnh giấy ban đầu và cặp số  $(d, e)$  hãy tìm phương án cắt hình vuông thỏa mãn các điều kiện trên mà có cạnh lớn nhất.

**Input:**

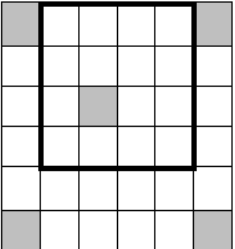
- Dòng đầu chứa năm số nguyên  $a, b, c, d, e$  ( $e \leq c \leq a \times b; 1 \leq d \leq \min\{a, b\}$ );
- Dòng thứ  $k$  trong  $c$  ( $1 \leq k \leq c$ ) dòng tiếp theo chứa hai số nguyên  $i_k, j_k$  mô tả ô đã tô màu.

**Output:** Ghi ra một số nguyên là độ dài cạnh hình vuông lớn nhất tìm được.

**Giới hạn:**

- Có 30% số test ứng với 30% số điểm của bài có  $a, b \leq 20$ ;
- Có 30% số test khác ứng với 30% số điểm của bài có  $a, b \leq 200$ ;
- Có 20% số test khác ứng với 20% số điểm của bài có  $a, b \leq 2000$  và  $c \leq 10^5$ ;
- Có 20% số test còn lại ứng với 20% số điểm của bài có  $a, b \leq 6000$  và  $c \leq 10^5$ .

**Ví dụ:**

Input	Output	Minh họa
6 6 5 2 2 1 1 1 6 3 3 6 1 6 6	4	 Một phương án chọn cắt hình vuông

**BÀI 2. KHÔI PHỤC DỮ LIỆU**

Ngân đang chuẩn bị chủ đề trình bày trong cuộc thi khoa học trẻ sắp diễn ra. Chủ đề về thuật toán khôi phục các giá trị bị mất của chuyển động khớp xương bàn tay trong chuỗi thời gian. Cụ thể, dữ liệu khớp xương gồm ba dãy giá trị  $A, B, C$  có cùng độ dài  $m$ , trên mỗi dãy các phần tử được đánh số từ 1 đến  $m$  từ đầu về cuối. Nhằm đánh giá được hiệu quả thuật toán khôi phục khớp xương, Ngân cần chọn ra các vị trí và đánh dấu mất mát trên dữ liệu để thử nghiệm. Tuy nhiên, Ngân thắc mắc có bao nhiêu cách chọn thỏa mãn:

- Có ít nhất một vị trí được chọn;
- Số lượng vị trí được chọn trên cả ba chuỗi chia hết cho  $k$  ( $0 < k \leq 100$ );
- Không tồn tại  $i$  ( $1 \leq i \leq m$ ) mà vị trí  $i$  trên cả ba đồng thời được chọn.

**Yêu cầu:** Gọi  $s$  là cách chọn thỏa mãn, hãy giúp Ngân tính  $s \% (10^9 + 7)$ .

**Dữ liệu:** Vào từ thiết bị vào chuẩn gồm một dòng chứa hai số nguyên dương  $m, k$ .

**Kết quả:** Ghi ra thiết bị ra chuẩn một số nguyên  $s \% (10^9 + 7)$  tính được.

**Ví dụ:**

Dữ liệu vào	Kết quả ra	Giải thích
2 4	9	Dưới đây là các cách chọn thỏa mãn, trong đó, số 1 thể hiện vị trí được chọn, ngược lại số 0 thể hiện vị trí không được chọn. <div>11   11   10   11   11   10   01   01   00</div> <div>11   10   11   01   00   01   11   10   11</div> <div>00   01   01   10   11   11   10   11   11</div>

**Giới hạn:**

- Subtask 1 (30% số điểm):  $m \leq 5$ ;
- Subtask 2 (40% số điểm):  $m \leq 5 \times 10^3$ ;
- Subtask 3 (30% số điểm):  $m \leq 5 \times 10^8$ .

## Bài 4: Cắt giấy

### Mảng PrefixSum

FOR  $i = 1 \rightarrow 5$

Xây dựng bằng công thức  $\text{PrefixSum}[i] = \text{PrefixSum}[i-1] + A[i]$

- $A[5] = \{10, 20, 30, 40, 50\}$
- Tính tổng  $A[L] + A[L+1] + \dots + A[R] = \text{PrefixSum}[R] - \text{PrefixSum}[L-1]$
- Trong độ phức tạp bằng  $O(1)$

$i$	1	2	3	4	5
A	10	20	30	40	50
PrefixSum	10	30	60	100	100+50 = 150

$$A[2] + A[3] + A[4] = \text{PrefixSum}[4] - \text{PrefixSum}[1] \\ = 100 - 10 = 90$$

14



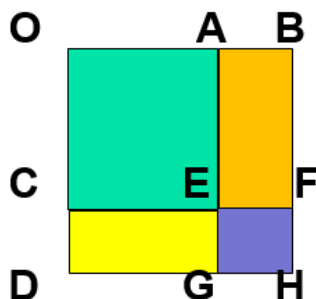
## Bài 4: Cắt giấy

### Mảng PrefixSum trong mảng 2 chiều

$\text{PreSum}[i][j] =$  tổng các phần tử của hình chữ nhật từ  $(1, 1) \rightarrow (i, j)$

### Xây dựng $\text{PreSum}[i][j]$ :

$\text{board}[i][j]$ ,  $\text{PreSum}[i-1][j-1]$ ,  $\text{PreSum}[i][j-1]$  và  $\text{PreSum}[i-1][j]$



$$\text{OBHD} = \text{OAEC} + \text{ABFE} + \text{EFGH} + \text{CEDG}$$

$$\text{OBHD} = \text{OAGD} + \text{OBFC} + \text{EFGH} - \text{OAEC}$$

Từ công thức thứ 2, có:  $\text{PreSum}[i][j]$   
 $= \text{PreSum}[i][j-1] + \text{PreSum}[i-1][j] +$   
 $\text{board}[i][j] - \text{PreSum}[i-1][j-1]$

15



```

int check(int i, int j, int r) {
    int u = i+r-1, v = j+r-1;
    int cnt = preSum[u][v] + preSum[i-1][j-1] - preSum[u][j-1] - preSum[i-1][v];
    return cnt <= e;
}

void solve() {
    for (int i = 1; i <= a; i++) {
        for (int j = 1; j <= b; j++) {
            preSum[i][j] = preSum[i-1][j] + preSum[i][j-1] + board[i][j] - preSum[i-1][j-1];
        }
    }

    for (int r = min(a,b); r >= 1; r--) {
        for (int i = 1; i <= a; i++) {
            for (int j = 1; j <= b; j++) {
                if (i+r-1 <= a && j+r-1 <= b && (r % d == 0)) {
                    if (check(i, j, r)) {
                        cout << r << endl;
                        return;
                    }
                }
            }
        }
    }
}

/// Cach 1: FOR r trau
/// Cach 2: r += d
/// Cach 3: Chat nhi phan de tim tmp --> r
/// Dpt: log (n) * n^2
for (int tmp = min(a,b)/d; tmp >= 1; tmp--) {
    int r = tmp*d;
}

```



## Bài 3: Khôi phục dữ liệu – Chuyên tin 2022

---

**Subtask 1:**  $m \leq 5 \rightarrow$  sinh tổ hợp

**Subtask 2:** Quy hoạch động, xây dựng công thức truy hồi  
Độ phức tạp:  $O(sz * m)$

**Subtask 3:** Speed up bằng nhân ma trận  
 $m \leq 5 \cdot 10^8 \rightarrow$  Không thể dùng FOR cho qhd thông thường.  
Độ phức tạp:  $O(sz^3 * \log(m))$  với  $sz$  = kích thước của ma trận



### Bài 3: Khôi phục dữ liệu – Chuyên tin 2022

Tối ưu bằng nhân ma  
trận:

Vector cột

$$\begin{array}{l} dp[i][0] \\ dp[i][1] \\ dp[i][2] \\ \dots \\ dp[i][k-1] \end{array} = A * \begin{array}{l} dp[i-1][0] \\ dp[i-1][1] \\ dp[i-1][2] \\ \dots \\ dp[i-1][k-1] \end{array}$$

Hàng đầu tiên của A:

1 0 0 ... 0 3 3

$$\begin{aligned} dp[i][0] &= dp[i-1][0] * \text{số cách chọn thêm 0 phần tử} \\ &\quad + (dp[i-1][1] * \text{số cách chọn thêm k-1 phần tử}) = 0 \\ &\quad + (dp[i-1][2] * \text{số cách chọn thêm k-2 phần tử}) = 0 \\ &\quad \dots \\ &\quad + (dp[i-1][k-2] * \text{số cách chọn thêm 2 phần tử}) \\ &\quad + (dp[i-1][k-1] * \text{số cách chọn thêm 1 phần tử}) \\ &= dp[i-1][0] + dp[i-1][k-2] * 3 + dp[i-1][k-1] * 3 \end{aligned}$$

Xây dựng công thức cho  $dp[i][1], dp[i][2], \dots, dp[i][k-1]$

$$ans = dp[m][0] - 1$$

