

Bài A. GENTEST

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Hạn chế

- Trong tất cả các test: $1 \leq n \leq 10^9$, $1 \leq m \leq 10^5$
- Có 10% số test với: $1 \leq n, m \leq 500$
- 30% test tiếp theo với: $1 \leq m \leq 500$
- 30% test tiếp theo với: $1 \leq n \leq 500$
- 30% test tiếp theo với ràng buộc gốc

Subtask 1: Làm theo đúng các bước đề bài yêu cầu

Subtask 2: Gọi E là tập tất cả các cạnh có thể có, thay vì xóa bớt cạnh trong E ta sẽ tính lại vị trí thật sự trên E của cạnh cần in (vị trí của cạnh đó nếu E không bị xóa). Vị trí thật sự i^* sẽ thỏa mãn: $i^* = i +$ số $j^* \leq i^*$ với j^* bị xóa trước i^* . Nếu cài đặt trực tiếp cách này sẽ có độ phức tạp $O(m^2)$

Subtask 3: Tương tự subtask 2, do $|E|$ nhỏ nên có thể dùng IT để quản lý và tính nhanh i^*

Subtask 4: $|E|$ quá lớn nên ta sẽ kiểm soát phần bù của E , cụ thể là dùng 1 cây trie (hoặc IT động) để quản lý tập những cạnh đã bị xóa. Thao tác tính i^* có thể tìm kiếm nhị phân và dùng cây quản lý này để kiểm tra ($O(m \log^2 n)$), hoặc tìm kiếm trực tiếp trên cây quản lý này ($O(m \log n)$)

Bài B. WOOD

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Hạn chế

Với mỗi test, nếu chỉ tính đúng chi phí cắt sẽ được 50% điểm của test đó, để nhận được 50% điểm này thì dòng thứ hai bạn cần để trống.

- Subtask 1 (20%): $k \leq 8$
- Subtask 2 (20%): $n = 1$
- Subtask 3 (20%): Thứ tự Hùng nhớ là một thứ tự hợp lý và tối ưu
- Subtask 4 (20%): Thứ tự Hùng nhớ là một thứ tự hợp lý
- Subtask 5 (20%): Ràng buộc gốc

Subtask 1: Duyệt hết tất cả các thứ tự cắt có thể có.

Subtask 2: Miếng gỗ có dạng 1 thanh dài, lúc này đưa về bài toán 1 chiều. Gọi độ dài sau khi cắt của các thanh gỗ là a_1, a_2, \dots, a_n , ta sẽ đảo ngược quá trình cắt: hợp nhất các thanh gỗ này lại thành 1 thanh duy nhất với chi phí hợp nhỏ nhất. Thuật toán là tham lam, sử dụng hàng đợi ưu tiên để tìm ra vị trí hợp nhỏ nhất.

Subtask 5: Sử dụng tư tưởng của subtask 2. Với một hình chữ nhật, liệt kê hết các phép cắt nằm trong hình đó và có thể chia đôi hình đó. Tập các lát cắt này sẽ chia hình chữ nhật thành nhiều thanh tương tự như subtask 2, do đó có thể tính toán tìm ra thứ tự cắt cho riêng tập này. Các hình chữ nhật phát sinh sau khi cắt sẽ trở thành các bài toán con, ta sẽ nạp vào stack để xử lý từng hình một. Bài toán có hai phần riêng biệt: Phần cấu trúc dữ liệu để lọc ra các lát cắt phù hợp với một hình, và phần nội hàm thuật toán là thuật tham lam.

Subtask 3: Bỏ qua phần nội hàm thuật toán, chỉ kiểm tra phần cấu trúc dữ liệu.

Subtask 4: Bao gồm phần nội hàm thuật toán, nhưng lược bỏ 1 kỹ thuật xử lý trong cấu trúc dữ liệu.