HƯỚNG DẪN GIẢI SUMMER CONTEST 01

Bài A: Làng Xì Trum mở hội

Đây thực chất là bài toán Josephus, ai sẽ là người cuối cùng ? Bạn không thể dùng cách bình thường bằng việc chạy vòng lặp for và xóa phần tử ở vị trí K lần lượt ở từng bước vì số N của bài toán khá lớn N^12 nếu bạn làm cách trên thì chắc chắn sẽ bị TLE. Bạn bắt buộc phải dùng thuật toán sau:

Có một bài toán đặt ra như sau: "Cho N người đứng thành vòng tròn và số M (ở đây cho dễ ta sẽ giả sử M < N). Bắt đầu từ người số 1, anh ta sẽ đếm 1, người bên phải anh ta đếm 2, … cho tới người đếm M sẽ tự động bước ra khỏi vòng tròn đó và người bên phải anh ta tiếp tục đếm lại 1 … Cứ thế cho tới khi vòng không còn người nào. Câu hỏi đặt ra là ai là người ra khỏi vòng cuối cùng ?"

Bằng một ít suy nghĩ có lẽ ta sẽ tìm ra ngay các trả lời cho câu hỏi này. Chẳng hạn, bạn có thể "giả lập" trò chơi trên với một dãy số nguyên từ 1 đến N và số M, thực hiện bỏ từng số cho đến khi nào dãy chỉ còn một số và đó chính là kết quả. Như vậy bạn phải chạy N "vòng" (vì mỗi "vòng" bỏ một số) và mỗi "vòng" bạn phải đếm qua M người. Như vậy với cách giả lập trò chơi bạn mất M x N thao tác đếm.

Thực chất lí do mà mình viết bài này vì … thực sự quá (?) sung sướng khi bắt gặp trên VNOI một cách giải mới có thể đưa số lần tính toán giảm xuống rất nhiều (MlogN). Ta thử ví dụ một trò chơi với N = 8 và M = 5 nhé:

Đầu tiên ta có:

1 2 3 4 ***5*** 6 7 8

Với những số không bị bỏ đi và nằm phía trước số bị bỏ, đem chúng cho vào phía sau (thì coi như trò chơi mới với dãy gồm N – 1 người)

6 7 8 9 ***10*** 11 12

Tiếp tục:

11 12 13 14 ***15*** 16

16 17 18 19 ***20***

21 22 23 24 ***25***

26 27 28 29 ***30***

…

Bạn sẽ chú ý tới 3 dòng cuối cùng (và một số dòng ở dưới đó). Hãy suy nghĩ một chút sẽ biết tại sao lại làm như vậy. Chỉ là hợp thức hóa vấn đề phát sinh này thôi, không có gì quan trọng cả. Điều quan trọng ở đây là, trong dãy số mới, các số bị bỏ đi bao giờ cũng là k x M và số bị bỏ đi cuối cùng chính là M x N. Ta sẽ tìm hiểu số mang thứ tự M x N trong dãy mới kia sẽ tương ứng với người mang thứ tự bao nhiêu ở dãy 1..N ban đầu, và đó sẽ là kết quả.

Một nhận xét, các số M x k luôn bị bỏ đi,và M x k + r (0 < r < M) sau một lần chuyển sẽ chuyển lên vị trí N + (M – 1) x k + r. Cho phép mình giải thích cái công thức kia một chút. Tại sao số M x k + r sẽ trở thành số N + (M – 1) x k + r sau một phép "chuyển ra sau" ? Giả sử bạn đang xóa số M x k, thì đương nhiên phía trước đó nhất định phải còn lại M – 1 số (xem ví dụ), bạn chuyển M – 1 số đó ra sau (đương nhiên là sau số N), như vậy bạn bỏ qua bao nhiêu lần M thì cũng chuyển bấy nhiêu lần M – 1 ra sau, đó là lí do tại sao ta có cái công thức đấy.Như vậy ta có vị trí cuối cùng là M x N và công thức chuyển vị trí (M x k + r) thành (N + (M – 1) x k + r), ta sẽ tìm cách chuyển về cho tới khi vị trí tính được nhỏ hơn N là xong.
Cách chuyển như thế nào ? Giả sử vị trí hiện tại là P, ta có:

$$P = N + (M – 1) \times k + r$$

Do đó:

$$P – n – 1 = (M – 1) \times k + (r – 1)$$

Suy ra:

$$k = [(P – n – 1) / (M – 1)] \text{ và } P + k – N = M \times k + r$$

Do đó vị trí trước sẽ là:

$$M \times k + r = P_{trước} = P_{sau} + [(P_{sau} – n – 1)/(M – 1)] – N$$

Tính vị trí trước như thế cho tới khi nào nó nhỏ hơn hoặc bằng N thì dừng. Đó chính là vị trí ban đầu của M x N.
(Bài viết được trích dẫn từ http://kienthuc24h.com)

Bài B: Tính diện tích

Đây là bài dễ nhất trong 4 bài ở kỳ thi này, đây là bài không cần dùng thuật toán. Đề bài yêu cầu bạn tính diện tích vùng màu đỏ, dễ dàng nhận thấy vùng màu đỏ chính là diện tích của hình vuông trừ diện tích của hình tròn. Do bạn đã có bán kính ban đầu đề bài đã cho nên bạn dùng bán kính này để làm 2 việc sau:

- Tính ra chiều dài cạnh của hình vuông (bán kính * 2). Có chiều dài cạnh thì sẽ tính được diện tích hình vuông.
- Sau đó tiếp tục dùng bán kính này để tính diện tích hình tròn. Chỉ có một điều lưu ý ở đây khiến nhiều bạn sai là bạn phải dùng công thức 2 * acos (0.0) để tính số PI (dùng 3.14 là không chính xác) và bạn phải in ra được 2 số phía sau dấu phẩy (xem thêm source code để hiểu cách in ra cho hợp lý).

Kết quả = diện tích hình vuông – diện tích hình tròn.

Bài C: Người Viking

Đọc đề bài chắc chắn bạn sẽ nghĩ liền ra được đây là bài toán tìm đường đi ngắn nhất. Chính vì vậy bạn tưởng tượng tất cả các đảo được xem là các đỉnh trên đồ thị. Do đề bài nói rằng "đoạn đường đi qua lại giữa các đảo đều bằng nhau". Điều đó có nghĩa là đây là đồ đồ thị vô hướng không có trọng số. Về thuật toán tìm được đi ngắn nhất có nhiều thuật, bài này bạn chỉ cần dùng thuật toán BFS là xử lý gọn gàng bài toán. Nếu chưa biết về thuật toán BFS bạn có thể tham khảo link bên dưới:

https://www.ics.uci.edu/~eppstein/161/960215.html

Một vấn đề lưu ý nữa là bài toán cho vào các đảo là tên, trong khi bạn học thuật toán tìm đường đi thì các đỉnh là số. Để quy bài toán từ tên về số bạn cần phải sử dụng thêm một cấu trúc dữ liệu khác là map của STL (Standard Template Library). Xem thêm cách sử dụng map tại đây:

http://www.cplusplus.com/reference/map/map/

Xem thêm **source code** để hiểu cách sử dụng map đơn giản.

Bài D: Trò chơi cờ lau

Khi bạn đọc 1 bài toán có liên quan đến việc gộp nhóm và một đối tượng có thuộc về một nhóm hay không thì thường bạn sẽ phải sử dụng cấu trúc dữ liệu Disjoin Set Union (DSU). Cấu trúc dữ liệu này dùng để tập hợp các phần tử lại với nhau thành một tập lớn. Cấu trúc này bao gồm 2 thao tác cơ bản Find (tìm một đối tượng có thuộc về tập hợp các đối tượng khác hay không). Union (Ghép 2 tập hợp lại với nhau).

Đây cũng là cấu trúc dữ liệu cơ bản bắt buộc bạn phải biết nếu như bạn nếu học lên cao hơn là thuật toán **Kruskal**. Bài toán này chính là bài toán yêu cầu bạn dùng cấu trúc dữ liệu DSU để gộp nhóm cho các cây cờ lau. Chi tiết về cách sử dụng Disjoin set các bạn có thể tham khảo tại đây:

https://www.topcoder.com/community/data-science/data-science-tutorials/disjoint-set-data-structures/

Bạn cần tham khảo thêm source code để hiểu thêm cấu trúc dữ liệu này.