

C

Bài này là bài A trên contest codeforces. Sáng anh đặt tên file PDF nhầm. Đây là bài nhẹ nhàng nhất trong 3 bài. Đề bài viết khá phức tạp, nhưng yêu cầu tìm tất lại như sau:

- Cho 4 số t_1, t_2, t_3, t_4 . Cần chọn t_5 nhỏ nhất có thể, sao cho tổng của 5 số trừ đi số nhỏ nhất và số lớn nhất phải không lớn hơn t .

Giả sử $t_1 \leq t_2 \leq t_3 \leq t_4$.

Các trường hợp có thể xảy ra:

- $sum(t_2, t_3, t_4) \leq t$. Trường hợp này các bạn thêm bất cứ số nào vào cũng được \rightarrow kết quả là infinite.
- $sum(t_1, t_2, t_3) > t$. Trường hợp này không có cách nào hết, vì có thêm 1 số nhỏ hơn t_1 vẫn không giải quyết được vấn đề.
- Các trường hợp còn lại thì duyệt trâu là được (do chỉ có chính xác 2 chữ số sau phần thập phân).

B

Bài B này là một bài không khó, nhưng cài đặt có phần phức tạp. Với mỗi truy vấn, chúng ta cần tìm đoạn dài nhất có trung bình cộng "độ dốc" lớn hơn hoặc bằng g .

Giả sử, chúng ta chỉ xét các đoạn nguyên (không tính phần thập, chỉ lấy nguyên 1 đoạn). Chúng ta sẽ tính a_i là độ dốc của đoạn thứ i .

Để đơn giản hơn, với truy vấn g , chúng ta sẽ trừ tất cả a_i đi g . Khi đó, bài toán quy về tìm một đoạn liên tiếp có tổng âm. Đây là 1 bài toán cơ bản, có rất nhiều cách giải quyết (chặt nhị phân, hoặc dùng two pointer cũng được). Anh sẽ không trình bày ở đây, nếu có vướng mắc thì tối mai anh sẽ giải đáp cho bọn em.

Tiếp theo, sau khi tìm được một số đoạn (l, r) có độ dài lớn nhất có thể, chúng ta cần tìm kết quả bài toán.

Nhận thấy chúng ta sẽ tìm cách thêm một phần của đoạn $(l-1)$ hoặc $(r+1)$ vào đoạn (l, r) để có đoạn dài nhất có thể. Lưu ý rằng chúng ta sẽ không bao giờ lấy cả 2 phần, mà chỉ lấy 1 nửa trái hoặc phải, do chúng ta sẽ luôn chọn phần có "độ dốc" lớn hơn, để đoạn đường thêm vào được là dài nhất có thể.

A

Bài này là bài C trên contest codeforces. Sáng anh đặt tên file PDF nhầm.

Đây là một bài xác suất khá khó.

Ý tưởng là chúng ta sẽ tính một mảng $ans[i]$ là xác suất xuất hiện của con quái vật thứ i . Với mỗi xác suất s_i ban đầu, chúng ta sẽ tìm cách tính xem s_i sẽ được "cộng" vào các $ans[j]$ nào. Dĩ nhiên, xác suất xuất hiện s_i chỉ có thể cộng vào k con quái vật nằm bên phải i (tính cả i).

Các bạn có thể xem đoạn mã bên dưới để dễ hình dung hơn:

```
1  int m = n - k;
2  for (int i = 0; i < n; i++) {
3      double spawnChance;
4      cin >> spawnChance;
5      double add = (k * 1.0) / n * spawnChance;
6
7      ans[i] += add;
8      for (int j = 1; j <= m; j++) {
9          add *= (m - j + 1.0) / (n - j);
10         ans[(i + j) % n] += add;
11     }
12 }
```

Đầu tiên, tỉ lệ quái vật i xuất hiện trong spawn pool cho mỗi pool là k/n . Do đó, $ans[i]$ sẽ được cộng bằng xác suất xuất hiện của quái vật i nhân với k/n .

Tiếp theo, để $ans[i+1]$ nhận được tỉ lệ xuất hiện của quái vật i , đồng nghĩa với $i+1$ phải xuất hiện và i không xuất hiện. Xác suất để quái vật i không xuất hiện là $(n-k)/(n-1)$ (Do quái vật i đã xuất hiện \rightarrow chỉ còn $n - 1$ quái vật, và có $n - k$ quái vật không được xuất hiện).

Tương tự, để $ans[i+2]$ nhận được tỉ lệ xuất hiện của quái vật $i \rightarrow i+1$ và i không xuất hiện. Xác suất để $i + 1$ không xuất hiện tương tự trên $(n-k)/(n-1)$, và xác suất để i không xuất hiện là $(n-k-1)/(n-2)$ (Do chỉ còn $n-2$ quái vật, và còn lại $n-k-1$ quái vật không xuất hiện).
