

BACTERIUM

Lời giải của SLoW MoTioN

Giả sử chọn ra m con vi khuẩn, con thứ i tỏa ra độc a_i và chịu được độc b_i , vậy để cả m con cùng sống thì: $b_i \geq \frac{\sum a_i}{m}$ với mọi i , hay: $\min(b_i) \geq \frac{\sum a_i}{m}$

Từ đây ta có ý tưởng: Giới hạn giá trị nhỏ nhất b_i và tìm số lượng vi khuẩn nhiều nhất có thể cùng sống sót với b_i là khả năng chịu độc nhỏ nhất.

Nhận xét: Khi có 1 tập các con vi khuẩn có thể chọn, và 1 giá trị trung bình lớn nhất có thể, cách tốt nhất để có nhiều vi khuẩn nhất là lần lượt chọn ra con tỏa ra ít độc nhất chừng nào còn thỏa mãn điều kiện. (Có thể được chứng minh bằng nhận xét khi chọn ra x số bất kì trong 1 tập S thì cách chọn có trung bình nhỏ nhất là chọn ra x số đầu tiên). Vậy đầu tiên thuật toán cần phải xác định giá trị trung bình lớn nhất có thể và tập vi khuẩn được phép chọn từ khi đó, sau đó phải tính toán được số lượng vi khuẩn nhiều nhất có thể được chọn ra.

I. Cách làm cơ bản cho ta độ phức tạp $n^2 \log_2(n)$:

- Duyệt mọi b_i , liệt kê ra tập tương ứng, sắp xếp và thử chọn.

II. Cách làm tốt hơn cho độ phức tạp $n(\log_2(n)^2)$:

- Vì khi $\min(b_i)$ giảm thì tập vi khuẩn ta có thể chọn để thỏa mãn mỗi con đều chịu độc tốt hơn $\min(b_i)$ được mở rộng, nên ta sẽ sort lại các con vi khuẩn theo độ chịu độc giảm dần để tiện xử lí. Quản lí một Segment Tree, nút i chứa đoạn từ $l \rightarrow r$ mang các thông tin tổng của các số trong đoạn này và số lượng các số trong đoạn này. Duyệt b_i giảm dần, với mỗi giá trị b_i phân biệt, ta thêm một giá trị a_j vào Segment Tree với mọi $b_j = b_i$. Sau khi thêm xong vào tập có thể chọn, ta tìm ra số lượng lớn nhất có thể chọn được khi đó bằng cách chặt nhị phân kết quả, và dùng segment tree để tính tổng của k (giá trị cần kiểm tra) số nhỏ nhất trong tập.
- Cách tính tổng trong $O(\log(n))$: Giả sử đang ở nút i và cần tính tổng k số nhỏ nhất trong các con của nó: Nếu i là một nút chỉ chứa 1 số thì có thể kết luận được luôn. Nếu nút con trái của i mà có số lượng phần tử là $g < k$ thì lấy tổng của chúng cộng vào kết quả, và đi vào nút con phải tính giá trị của $k - g$ số nhỏ nhất trong đó. Nếu không thì số cuối cùng chọn được sẽ nằm trong nút con trái nên ta chỉ cần đi vào nút con trái. Độ phức tạp việc này là $O(\log(n))$ vì chỉ phải đi vào nút con tối đa $\log(n)$ lần.

III. Cách làm có độ phức tạp $O(n \log(n))$:

- Cũng quản lí một Segment Tree như trong cách làm trên, tuy nhiên ta không cần phải chặt nhị phân độ dài mà có thể tính trực tiếp luôn dựa vào nhận xét: một tập đang có trung bình cộng là x , thêm vào một số $y \geq x$ thì trung bình cộng của tập đó chắc chắn không giảm và không vượt quá y .
- Cách làm: Giả sử đang ở nút i và đã chọn được k số, có tổng bằng s , có trung bình lớn nhất có thể là b : Nếu i là một nút chỉ chứa 1 số thì có thể kết luận được luôn: Giá trị số đó mà

Free Contest 69

không lớn hơn b thì dựa vào nhận xét, ta lấy hết trong nút. Nếu không thì có thể giải bất phương trình bậc nhất để tìm được số lượng có thể lấy. Nếu không, thử thêm cả tập trong nút con trái vào tập đã chọn được, nếu thoả mãn thì lấy cả tập và tiếp tục chọn trong nút con phải, nếu không thoả mãn thì đi vào nút con trái. Điều này có thể chứng minh như sau: Nếu không chọn hết được nút con trái, thì trung bình cộng khi này là một số $x > b$, mà số x này lại bé hơn mọi số trong nút con phải, nên chọn thêm thì không thể giảm được. Độ phức tạp việc này cũng là $O(\log(n))$ với nhận xét như trên.
