

Dễ thấy chúng ta chỉ cần quan tâm số lượng team vượt team 1 chứ không cần quan tâm về mọi team còn lại như thế nào.

Vậy ta sẽ tìm cách để có thể lưu các team có thứ hạng cao hơn 1 ở đây ta có thể sử dụng bất kỳ nhưng để tối ưu chúng ta sẽ sử dụng `unordered_set`.

Với mỗi team khi mà họ được công bố là làm được 1 bài nữa thì họ hoặc là có thứ hạng cao hơn 1 hoặc là có thứ hạng vẫn bằng hoặc thấp hơn 1.

Vậy nếu team đó có kết quả cao hơn team 1 thì chúng ta sẽ thêm team đó vào danh sách các team có vị trí cao hơn 1.

Khi team 1 được cập nhật thì chúng ta kiểm tra lại những team có kết quả tệ hơn team 1 trong danh sách các team có thứ hạng cao hơn team 1.

Độ phức tạp: $O(N \times M)$.

Nhưng chúng ta cần thấy tại đây N thực tế là số lượng phần tử trong danh sách các team có thứ hạng cao hơn 1.

Vậy trường hợp tệ nhất là 1 là team ở vị trí cuối cùng rồi chúng ta tiến hành cho team 1 trở lại vị trí đầu bảng xếp hạng.

Nhưng nhiều nhất thì chúng ta chỉ có phải thực hiện 2 thao tác với mọi nhóm.

Vậy chúng ta vẫn có thể chạy được nó trong 1 giây giới hạn.

[Solution mẫu](#)