

Omówienie zadań

Filip Konieczny

3 listopada 2023

1 Dobry zawodnik

Pierwszą obserwacją jest fakt, że taki dobry zawodnik będzie zawsze istniał w turnieju. Jest wiele różnych dowodów tego faktu i niektóre z nich można przetłumaczyć na implementację algorytmu znajdującego tego zawodnika. Część z nich ma charakter indukcyjny, ale te oparte na metodzie maksimum okazują się proste w implementacji, dlatego taki właśnie omówimy. Okazuje się, że zachodzi następująca implikacja:

Obserwacja 1.1. Jeśli zawodnik ma co najmniej tyle zwycięstw co każdy inny zawodnik, to jest on dobry.

Dowód. Weźmy takiego zawodnika Z i powiedzmy, że wygrał on k razy. Weźmy jakiegoś zawodnika z którym przegrał P . Powiedzmy, że Z nie wygrał z P pośrednio. Wtedy P wygrał z każdym zawodnikiem z którym wygrał Z ORAZ wygrał z Z , czyli wygrał co najmniej $k + 1$ razy, sprzeczność. \square

1.1 Implementacja

Dla każdego zawodnika wyznaczyć liczbę jego wygranych meczy i wypisać tego, który ma ich najwięcej (w przypadku remisów można wypisać dowolnego z największą liczbą zwycięstw).

I w sumie tyle.

2 Gra w dzielniki

Rozpiszcie sobie strategię wygrywającą dla małych n (na przykład dla mniejszych niż 10). Powinien się pokazać jakiś wzorzec, który pomoże w rozwiązaniu zadania.

Bonusowe punkty satysfakcji za dowód, że zaobserwowany wzorzec istotnie działa dla dowolnego n .

3 Przycisk

Rozwiązanie, które za każdym razem, gdy klikany jest przycisk o numerze $n + 1$ leci po całej tablicy i ustawia wszystkie wartości na poprawną wartość nie przejdzie ze względu na ograniczenia czasowe (wyobraźcie sobie test, gdzie jest maksymalny rozmiar tablicy i maksymalna liczba zapytań, i każde z nich to naciśnięcie przycisku $n + 1$).

Okazuje się, że nie trzeba być aż tak skrupulatnym, można do tego problemu podejść „leniwie”. Co to znaczy? Trzymamy sobie pomocniczą zmienną z aktualną maksymalną wartością w tablicy (taką wartość łatwo zaktualizować niezależnie od tego, który przycisk został kliknięty).

Teraz w momencie kliknięcia przycisku o numerze $n + 1$ (niech aktualne maksimum to M) zamiast aktualizować całą tablicę zapamiętujemy sobie (np w zmiennej lul), że wszystkie liczby mniejsze niż M są teraz równe M .

Jak teraz obsłużyć kolejne zapytanie? Jeśli będziemy musieli zwiększyć wartość którejś pozycji, to odczytujemy ją i jeśli jest mniejsza niż lv , to ustawiamy ją na lv i dopiero wtedy zwiększamy. Jeśli była większa bądź równa, to po prostu zwiększamy.

Na końcu wypisujemy tablicę, pamiętając o tym, że jeśli przy wypisywaniu znajdziemy liczbę mniejszą niż lv , to należy zmienić jej wartość na lv .