

3C – Sumy

autor zadania: Mateusz Radecki

Treść

Dany jest ciąg wag ryb. Jedna ryba może zjeść inną tylko, gdy jest od niej ściśle cięższa. Po zjedzeniu jej waga rośnie o wagę zjedzonej ryby, a zjedzona ryba znika. Dla każdej ryby pytamy, czy może w stawie nastąpić ciąg zjedzeń, po którym pozostanie ona ostatnią, niezjedzoną rybą.

Rozwiązanie

Po pierwsze zauważmy, że jeśli jakaś ryba może pozostać ostatnią rybą po jakimś ciągu zjedzeń, to równie dobrze ona mogła zjeść każdą inną rybę. Powiedzmy, że istnieją ryby A , B i C o wagach odpowiednio a , b i c . Jeśli B zjadła C , a potem A zjadła B , to musiało zachodzić $b > c$ oraz $a > b + c$. Łatwo zauważyć, że w takim razie musiało też zachodzić $a > b$ oraz $a + b > c$, czyli A mogła zjeść też bezpośrednio C .

Teraz zauważmy, że jeśli jakaś ryba mogła pozostać ostatnia, to wszystkie ryby cięższe od niej również mogą pozostać ostatnie. Powiedzmy, że znamy dwie ryby A oraz B i zachodzi $A \geq B$. Za każdym razem, gdy B miałaby zjeść jakąś inną rybę, zamiast tego A mogłaby ją zjeść. W momencie, gdy B zjadłaby A , to zamiast tego A może zjeść B .

Daje nam to informację, że jeśli posortujemy ryby niemalejąco po wadze, to ryby na pewnym sufiksie mogą pozostać ostatnie. Pozwala nam to użyć wyszukiwania binarnego, by mnożąc złożoność przez $\log(n)$ sprowadzić zadanie do sprawdzenia, czy pewna konkretna ryba może pozostać ostatnia.

Pozostaje jedynie pytanie w jakiej kolejności sprawdzana ryba powinna próbować zjadać pozostałe. Jeśli może zjeść którąkolwiek z nich, to na pewno może zjeść najlżejszą, co tylko podniesie jej wagę i być może pozwoli zjeść większe ryby. Skoro i tak posortowaliśmy ciąg, to przejdźmy się po nim od najlżejszych ryb próbując po kolei je jeść. Jeśli któregoś zjeść nie możemy, to znaczy, że sprawdzana ryba nie może pozostać ostatnia.

Cały algorytm, zarówno przez sortowanie, jak i wyszukiwanie binarne, ma złożoność $\mathcal{O}(n \cdot \log(n))$.