

Zadanie: PER

Permutacja 2



Warsztaty ILO 2017-2018, grupa olimpijska.

Zadanie jest trochę inne niż zwykłe zadania. Polega na napisaniu backtracka i wymyśleniu pewnych tricków żeby go przyspieszyć. Na pewno jest wiele dróg, żeby je rozwiązać, oto jedna z nich.

Rozwiązanie wzorcowe

Najpierw puszcza się sito, żeby móc w czasie stałym sprawdzać pierwszość liczby.

Pierwszą próbą jest napisanie prostego backtracka, który po prostu sprawdza kolejne najmniejsze niewybrane liczby, znajduje taką, że jej suma z poprzednią liczbą jest pierwsza i wywołuje się rekurencyjnie, sprawdzając ewentualnie większe liczby.

Okazuje się, że taki backtrack po uruchomieniu działa około 3s. Jest to niezły i obiecujący wynik, ale niestety trzeba to jeszcze trochę przyspieszyć.

Można pomyśleć, że backtrack będzie miał problemy pod koniec, bo wtedy brakuje już liczb do sumowania żeby wyszła liczba pierwsza. Po sprawdzeniu okazuje się, że, rzeczywiście, pierwsza liczba która nie jest wybrana zachłannie (przy której był nawrót) to 30 liczba od końca wynikowego ciągu. To już daje nam szybkie rozwiązanie – możemy zachłannie obliczyć cały wynikowy ciąg bez 30 ostatnich wyrazów a 30 ostatnich wkleić ręcznie do kodu.

Ale pójdźmy dalej. Wiemy, że koniecznym warunkiem aby się dało, ustawić resztę ciągu, jest to, żeby każda nieustawiona liczba mogła zostać sparowana z dwoma innymi liczbami tak, żeby ich suma była liczbą pierwszą. Dodajmy więc sprawdzenie, czy dla każdej liczby istnieją przynajmniej dwie takie liczby (dorzucając pierwszą i ostatnio dostawioną liczbę). Jeśli nie istnieją możemy zawrócić backtrackiem i oszczędzić zbędnych obliczeń.

Po tej optymalizacji czas działania spadł do 0.35s, co jest wystarczające do zmieszczenia się w limitach czasowych.