11 Punktów

Aby uzyskać 11 punktów wystarczy przesłać pierwsze 2^{11} bitów z jednego ciągu do drugiego i policzyć wynik w tym drugim.

Binary search po wyniku

Użyjemy wyszukiwania binarnego po wyniku. Dla danego x chcemy stwierdzić czy istnieje co najmniej n elementów o wartości $\leq x$. W tym celu program każdy program wyśle do drugiego ilość elementów nie większych niż x w swoim zbiorze. Tutaj należy pamiętać, żeby jeden program najpierw wysyłał potem czytał, a drugi na odwrót.

Iteracji bin searcha będzie $\log_2 M = 21$, a w każdej iteracji obydwa programy wysyłają po $\log_2 M = 21$ bitów. Mamy zatem rozwiązanie, które wysyła łącznie $21 * (21 + 21) = 882$ bitów, co daje nam 35 punktów.

Optymalizacja 1

Po tym jak program A wyśle do programu H ilość elementów nie większych niż x w swoim zbiorze, program H może od razu porównać tą wartość zwiększoną o jego wartość z n i zmienić swój przedział w bin searchu. Następnie może odesłać programowi A jeden bit, w zależności od tego, czy ich suma jest większa czy mniejsza od n.

W ten sposób podczas każdej iteracji wysyłane są $21 + 1 = 22$ bity. To rozwiązanie wysyła łącznie $21 * (21 + 1) = 462$ bity, co daje nam 47 punktów.

Optymalizacja 2

Program A zamiast wysyłać łącznej liczby elementów nie większych od x może wysyłać ilość elementów na przedziale $[l, x]$, gdzie l to lewy koniec wyszukiwania binarnego. Program H musi utrzymywać zatem ilość elementów w zbiorze A na przedziałe $[0, l - 1]$. W momencie kiedy ustawia on w wyszukiwaniu binarnym $l = x + 1$, to dodaje do tej sumy ostatnio wysłaną liczbę przez program A. Dzięki temu w każdej kolejnej iteracji możemy wysłać o 1 bit mniej niż w poprzedniej. Wyślemy $21 + 20 + \ldots + 1 = 231$ bitów. Od A do H oraz 21 bitów od H do A, co łącznie da nam 252 bity i 53 punkty.

Rozwiązanie wzorcowe

Do tej pory rozwiązywaliśmy zadanie w złożoności $O(\log^2 M)$. Spróbujmy teraz zrobić to w $O(\log)$. Robimy wyszukiwanie binarne po wyniku, chcemy wysłać po jednym bicie od każdego programu. Nasuwa się tutaj wysłanie informacji, czy mamy co najmniej $\frac{n}{2}$ elementów nie wiekszych niż x. W momencie kiedy obydwa programy wyślą 0 wiemy, że wynik będzie wiekszy niż x. Analogicznie jeśli oba wyślą 1, to wynik będzie nie większy niż x. Problem pojawia się, gdy jeden program wyśle 0, a drugi 1. Możemy jednak zauważyć, że wynik będzie większy niż $\frac{n}{2}$-ga liczba w programie, który wysłał 1. Możemy zatem podzielić n przez 2, usunąc $\frac{n}{2}$ najmniejszych liczb z tego programu, a następnie kontynuować nasze wyszukiwanie binarne.

W każdej iteracji podzielimy n lub długość wyszukiwanego przedziału przez 2, więc iteracji będziemy mieli $2 \log_2 M$. Mamy zatem rozwiązanie, które wysyła około 84 bity i spokojnie mieści się w limicie 90 bitów na 100 punktów.