2A – Papier, kamień, nożyce

Autor zadania: Adam Gąsienica-Samek

Autor omówienia: Adam Gąsienica-Samek

Zadanie

Dane są dwa procesy, Algosia i Bajtek; każdy z nich posiada ciąg binarny długości $n = 5000$. Procesy mogą porozumiewać się między sobą grając wiele rund w grę papier, kamień, nożyce, przy czym w żadnym momencie jeden proces nie może mieć przewagi więcej niż jednego zwycięstwa. Należy zaimplementować strategię, za pomocą której procesy wymienią się swoimi napisami. Strategia powinna minimalizować liczbę odbytych rund.

Rozwiązanie $m \leq 10000$

Algosia i Bajtek chcą wysyłać między sobą bity, więc mogą do tego wykorzystać dwa ruchy na przykład papier odpowiadający 0 i kamień odpowiadający 1. Jeśli wyślą najpierw swój bit, a potem przeciwny, to niezależnie od tego jaki bit miał przeciwnik po dwóch rundach będzie remis. Jest to jedna z najprostszych strategii pozwalających zdobyć 3 punkty.

Rozwiązanie $m \leq 8750$

Czasami zdarza się, że już po pierwszej rundzie jest remis i procesy nie muszą wykonywać drugiego ruchu. Taka strategia działa dobrze na losowych testach, jednak nie radzi sobie, gdy ciągi są różne w każdym miejscu. Wylosujmy sobie dwa ciągi binarne długości n. Algosia po wczytaniu wejścia może zxorować swój ciąg z pierwszym, a Bajtek z drugim. Następnie po poznaniu wzajemnie swoich ciągów Algosia zxoruje wynikowy ciąg z drugim wylosowanym ciągiem, zaś Bajtek z pierwszym. Istotne jest tutaj, aby dwa procesy były w stanie wygenerować ten sam losowy ciąg, można to osiągnąć np. poprzez ustalenie tego samego stałego seeda w generatorze liczb losowych. Pozwala nam to teraz szukać strategii przy założeniu, że ciągi są losowe. Mamy szansę $\frac{1}{2}$, że bity Bajtka i Algosi będą równe, i $\frac{1}{2}$, że będą różne. Oczekiwana ilość odbytych rund to zatem $(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2) \cdot n = \frac{3}{2} \cdot n = 7500$.

Rozwiązanie $m \leq 5500$

W naszej grze mamy do dyspozycji trzy ruchy zamiast dwóch, narzuca się zatem, abyśmy spróbowali rozwiązać to korzystając z systemu trójkowego. W opisanej poniżej strategii Algosia będzie wiedziała, że liczba Bajtka mieści się w przedziale $[l_b, r_b)$ (gdzie $0 \leq l_b \leq r_b \leq 2^n$), analogicznie Bajtek będzie wiedział, że wynik Algosi leży w przedziale $[l_a, r_a)$. Zaczynamy od $l_a = l_b = 0$ i $r_a = r_b = 2^n$. Podczas ruchu każdy z graczy dzieli swój przedział na trzy równe części i wybiera ruch na podstawie tego, w której części faktycznie znajduje się jego liczba. Mamy $\frac{1}{3}$ szansy, że procesy wykonały ten sam ruch. Oczekiwana liczba odbytych rund to zatem $(\frac{1}{3} \cdot 1 + \frac{2}{3} \cdot 2) \cdot n \cdot \log_3 2 = \frac{5}{3} \cdot n \cdot \log_3 2 \approx 5258$. Warto zauważyć, że rozwiązanie to wymaga arytmetyki na bardzo długich liczbach, a konkretnie dodawania i odejmowania takich liczb oraz dzielenia ich przez 3. Pojedyncza taka operacja zajmuje czas $O(n)$, więc całość działa w $O(n^2)$.

Rozwiązanie $m \leq 5250$

W poprzednim rozwiązaniu wykonujemy wiele ruchów służących tylko temu, aby wyrównać wynik. Możemy spróbować w bardziej efektywny sposób wykorzystać te ruchy. W sytuacji przewagi, jedyna możliwość, która nie wprowadza ryzyka zakończenia rozgrywki, jest taka, że jeden proces wykonuje stały ruch, powiedzmy kamień, a drugi proces wykonuje jeden z dwóch ruchów: jeśli akurat prowadzi, to kamień lub nożyce, a jeśli jest do tyłu, to kamień lub papier. Proces, którego przedział jest w momencie ruchu większy będzie przesyłał informacje, a ten drugi będzie dawał stały ruch. W stanie równowagi w jednym ruchu przesyłamy $\log_2 3$ par bitów, a w stanie przewagi $\frac{1}{2}$ pary bitów. W stanie równowagi pozostajemy z prawdopodobieństwem $\frac{1}{3}$, a w stanie przewagi z prawdopodobieństwem $x = \frac{1}{2}$, co pozwala nam obliczyć, że średnio przez $\frac{3}{7}$ czasu będziemy w stanie równowagi. Oczekiwana liczba odbytych rund to zatem $n/(\frac{3}{7} \cdot \log_2 3 + \frac{4}{7} \cdot \frac{1}{2}) \approx 5181$.

Rozwiązanie $m \leq 5000$

Równowaga jest dla nas stanem korzystniejszym od przewagi, bo pozwala nam na przesłanie większej informacji. W stanie przewagi chcielibyśmy więc nie tylko coś przesłać, ale też przy okazji jak najbardziej dążyć do przejścia w stan równowagi. Wybierzmy sobie jakieś $x \leq \frac{1}{2}$. Dla $d = r - l$ podzielmy przedział $[l, r)$ na dwie nierówne części $[l, l + x \cdot d)$ oraz $[l + x \cdot d, r)$. Jeśli liczba procesu mieści się w mniejszej części, to wysyła on kamień zostawiając przewagę; w przeciwnym przypadku wraca do równowagi. W stanie przewagi pozostajemy wówczas z prawdopodobieństwem x, a przesyłamy z niego $\frac{1}{2} \cdot (-x \log_2 x - (1-x) \log_2 (1-x))$ par bitów (patrz: entropia). Obliczenia mówią, że w stanie równowagi będziemy przez $1 - \frac{2}{5-3x}$ czasu, zatem liczba par bitów przekazanych w jednej rundzie to

$$\left(1 - \frac{2}{5-3x}\right) \cdot \log_2 3 + \frac{1}{5-3x} \cdot \big(- x \log_2 x - (1-x) \log_2 (1-x)\big)$$

Optymalny x jest w okolicach 0.239788, prowadząc do wartości oczekiwanej na liczbę ruchów około 4853.98. Tak precyzyjna wartość x jest co prawda dość trudna w obsłudze, jednak dla $x = \frac{1}{4}$ wartość oczekiwana to około 4854.43, a dla $x = \frac{1}{3}$ około 4892.10. Obie te wartości pozwalały, aby zdobyć komplet punktów.