Ustawianie gońca (szkic rozwiązania)

Olimpiada Informatyczna Juniorów

Pola szachownicy możemy podzielić na przekątne (według wartości $x - y$) i antyprzekątne (według wartości $x + y$). Dla każdej przekątnej i antyprzekątnej możemy obliczyć ile jest tam interesujących pól.

Na początku zauważamy, że każdy goniec może atakować albo pola białe (o parzystej wartości $x-y$ oraz $x+y$) albo pola czarne (o nieparzystej wartości różnicy lub sumy współrzędnych). Rozpatrujemy oba przypadki osobno. Dalej opisujemy jak rozpatrzyć jeden z nich.

Niech R_1 oznacza największą liczbę interesujących pól leżących na pewnej przekątnej, a R_2 oznacza największą liczbę interesujących pól leżących na pewnej antyprzekątnej. Optymalnie ustawiony goniec nie może atakować więcej niż $R_1 + R_2$ interesujących pól (każdy goniec atakuje jedną przekątną i jedną antyprzekątną). Należałoby go ustawić na przecięciu optymalnej przekątnej i antyprzekątnej – jeżeli nie ma tam interesującego pola to jest to optymalne rozwiązanie. W przeciwnym razie okaże się, że tak ustawiony goniec atakuje tylko $R_1 + R_2 - 2$ interesujące pola – o co najwyżej dwa pola mniej niż ewentualne inne optymalne rozwiązanie (dwukrotnie niepotrzebnie policzyliśmy pole, na którym ustawiliśmy gońca).

Kandydatami na rozwiązania z wynikiem $R_1 + R_2$ są jedynie inne pary optymalnych przekątnych w wynikiem R_1 i antyprzekątnych z wynikiem R_2. Jeżeli którakolwiek z tych par nie ustawia gońca w interesującym polu to jest to rozwiązanie zadania, wypisujemy je i kończymy program.

Kandydatami na rozwiązania z wynikiem $R_1 + R_2 - 1$ są jedynie inne pary:

- przekątnych zawierających R_1 interesujących pól i antyprzekątnych zawierających $R_2 - 1$ interesujących pól,
- oraz przekątnych zawierających $R_1 - 1$ interesujących pól i antyprzekątnych zawierających R_2 interesujących pól.

Jak poprzednio, jeśli którakolwiek z tych par nie generuje gońca na interesującym polu to jest to rozwiązanie zadania.

Jeśli wszystko zawiodło, wypisujemy pierwszą opisaną wcześniej pozycję gońca stojącego na interesującym polu wynikającą z optymalnej przekątnej i antyprzekątnej.

Mogłoby się wydawać, że takie rozwiązanie będzie wolne, bo w pesymistycznym przypadku sprawdzanych par przekątna–antyprzekątna będzie dużo. Z każdej takiej pary wynika jednak inna pozycja gońca, a skoro jest tylko N różnych interesujących pól to, z zasady szufladkowej, po sprawdzeniu co najmniej $N + 1$ par, znajdziemy rozwiązanie zadania i przerwiemy program. Kluczowe jest jednak efektywne utrzymanie zbioru interesujących pól (np. w strukturze `unordered_set` z odpowiednią funkcją haszującą dla C++ lub `dict` w Pythonie lub, odrobinę wolniej, z użyciem wyszukiwania binarnego po posortowanej kolekcji par z wejścia).

Olimpiada finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025”.