

Zadanie: TET Tetris

Finał, plik źródłowy `tet.*`, dostępna pamięć 32 MB

13.05.2006

W grze Tetris dwuwymiarowe klocki spadają pionowo w dół na platformę. Dany klocek spada, dopóki nie napotka na przeszkodę w postaci innego klocka bądź samej platformy — wtedy zatrzymuje się i pozostaje na tej pozycji do końca gry. W tym zadaniu dla uproszczenia pomijamy znikanie całkowicie wypełnionych rzędów, które występuje w standardowej wersji gry. Ponadto założymy, że klocki mają wyłącznie postać poziomych pasków o krótszym (pionowym) boku długości jeden i że nie możemy manipulować w locie spadającym paskiem (obracać ani przesuwać).

Jedyne, co możemy zrobić, to wybrać kolejność spadania pasków. Twoim zadaniem jest, znając rozmiary wszystkich pasków oraz tory ich lotu, takie dobranie kolejności ich spadania, aby figura utworzona w wyniku ich spadnięcia była jak najniższa.

Zadanie

Napisz program który:

- wczyta ze standardowego wejścia opisy pasków i torów ich lotu,
- znajdzie taką kolejność ich spadania, aby wysokość figury utworzonej po ich spadnięciu była najmniejsza możliwa,
- wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 100\,000$), oznaczająca liczbę pasków, które spadną na platformę. Kolejnych n wierszy zawiera po dwie liczby całkowite l_i i p_i ($1 \leq l_i, p_i \leq 1\,000\,000\,000$), oddzielone pojedynczym odstępem i oznaczające odpowiednio długość i -tego paska i jego przesunięcie w poziomie względem początku platformy.

Wyjście

W pierwszym wierszu wyjścia powinna znajdować się jedna liczba całkowita, oznaczająca najmniejszą możliwą wysokość figury powstałej po spadnięciu wszystkich pasków w pewnej kolejności. Kolejnych n wierszy powinno zawierać opis przykładowej kolejności spadania klocków, realizującej to minimum. W $(i+1)$ -szym wierszu wyjścia powinna znajdować się jedna liczba całkowita, oznaczająca numer paska, który powinien spaść jako i -ty. Paski numerujemy liczbami od 1 do n w kolejności ich występowania na wejściu.

Jeżeli istnieje więcej niż jedna poprawna kolejność realizująca minimum wysokości, Twój program powinien wypisać którąkolwiek z nich.

Przykład

Dla danych wejściowych:

5
4 2
3 1
3 3
4 6
4 5

poprawnym wynikiem jest:

3
1
4
5
2
3