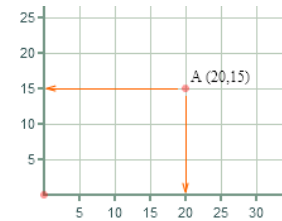


Przykrycie punktów



Wskazówki do rozwiązania zadania

Należy zauważyć, że prosta, w której zawiera się przeciwprostokątna szukanego trójkąta, ma równanie $x + y = M$, gdzie M jest pewną liczbą naturalną. Krótszy bok tego trójkąta ma właśnie długość M . Zatem dla każdego punktu (x_i, y_i) obliczamy wartość $M_i = x_i + y_i$. Na koniec wypisujemy największą wartość: $M = \max\{M_1, M_2, \dots, M_n\}$.

Poszukiwanie największej wartości M_i można zrealizować „w locie”, podczas wczytywania danych. Na początku przyjmujemy $M = 0$, a potem po wczytaniu współrzędnych kolejnego punktu podstawiamy $M := \max(M, x_i + y_i)$.

Kod programu w języku C++

```
#include <iostream>
using namespace std;

int main()
{
    int n, M = 0;
    cin >> n;
    while(n--)
    {
        int x, y;
        cin >> x >> y;
        M = max(M, x + y);
    }
    cout << M << endl;
    return 0;
}
```

Przy poszukiwaniu największej wartości wyrażenia M przyjmujemy jego początkową wartość równą zero, gdyż wszelkie inne wartości będą na pewno dodatnie.

Uwagi

Przy obliczaniu wyrażeń całkowitych należy zwracać baczną uwagę na możliwość przekroczenia zakresu typu `int`. W tym zadaniu to nigdy nie zajdzie, ponieważ współrzędne punktów mają wartość co najwyżej 10^6 , a maksymalna wartość w typie `int` ze znakiem przekracza $2 \cdot 10^9$.