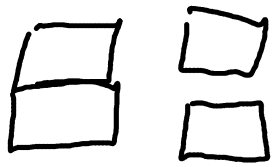
K23 - Zamiatanie

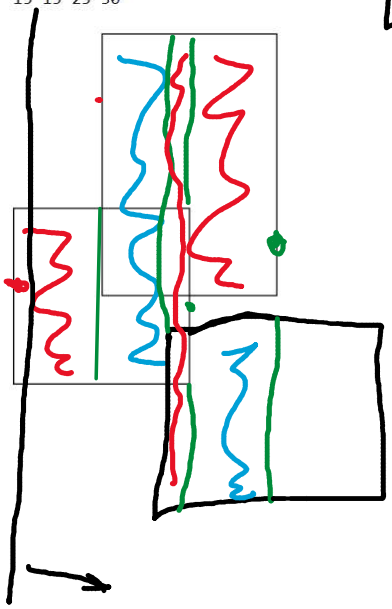
23 May 2025 17:59

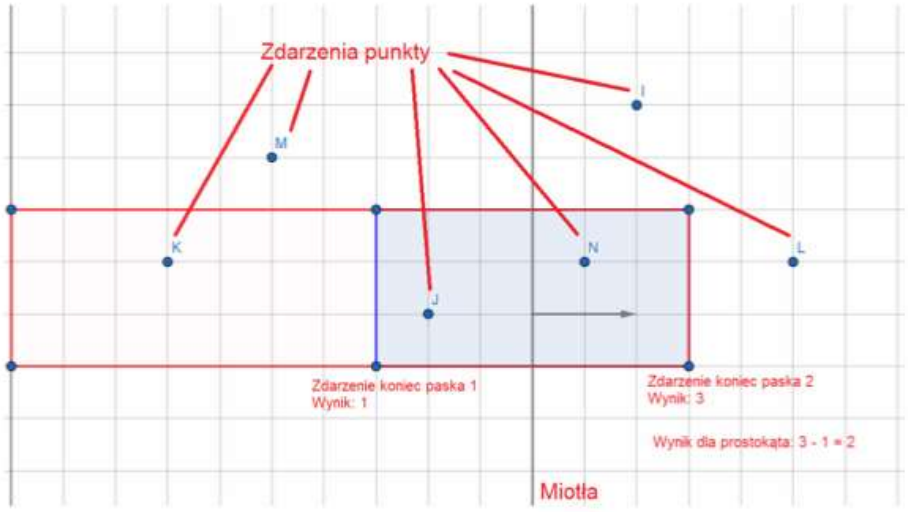
Przykład

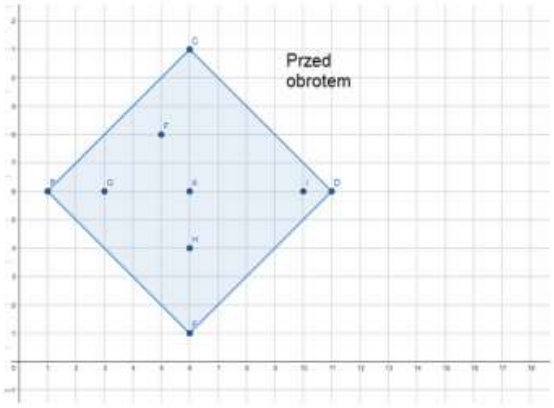
Dla danych wejściowych:

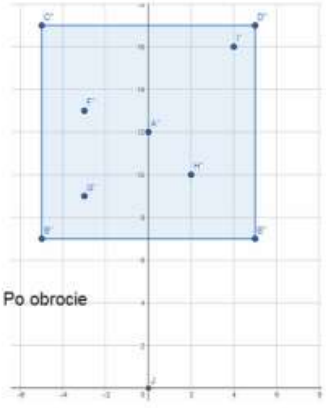
```
2
10 10 20 20
15 15 25 30
```












```
18      if(v[node] > 0)
17          sum[node] = maxa-mina+1;
16      else if(mina < maxa)
15          sum[node] = sum[node*2]+sum[node*2+1];
14      else
13          sum[node] = 0;
```

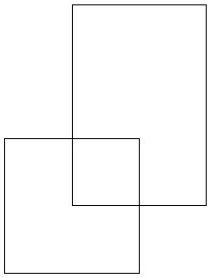
Przykład

Dla danych wejściowych:

```
2
10 10 20 20
15 15 25 30
```

```
struct Edge
{
    int x, y0, y1;
    bool isStart;
```

```
2
10 10 20 20
15 15 25 30
```

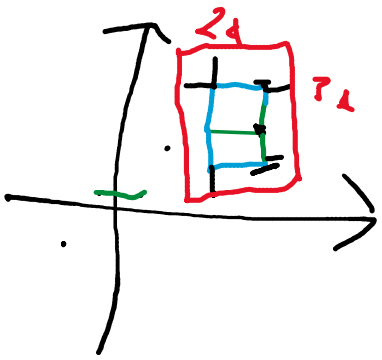


```cpp
struct Edge
{
    int x, y0, y1;
    bool isStart;

    friend bool operator < (const Edge& a, const Edge& b)
    {
        if(a.x == b.x)
            return a.isStart < b.isStart;
        return a.x < b.x;
    }
};
```

poprawną odpowiedzią jest:

```
225
```



$\Delta < curr.x - q.front().x$

Algorytm krok po kroku

Nasz algorytm zatem wygląda tak:

Posortuj punkty po x i przeglądaj w tej kolejności. Dla każdego punktu:

1) Usuń z początku kolejki (i jednocześnie z seta) wszystkie punkty, których $x < x_{akt} - \Delta$.

2) Sprawdź punkty na secie między lowerboundem $y_{akt} - \Delta$, a lowerboundem $y_{akt} + \Delta$ i zaktualizuj Δ (dotychczasowy najlepszy wynik).

3) Dodaj nowy punkt na kolejkę oraz dodaj na seta parę (y_{akt}, akt)

$k \cdot \pi \cdot \left(\frac{\Delta}{2}\right)^2 < 3\Delta \cdot 2\Delta$

$O(n \log n)$

$k \cdot \pi \cdot \frac{\Delta^2}{4} < 6\Delta^2$

$k\pi\Delta^2 < 24\Delta^2$

$k\pi < 24$

$k < 8$