

## Przecinające się odcinki – opracowanie wyników

3) Struktury zdarzeń i stanu:

- Struktura stanu została zaimplementowana jako kolejka priorytetowa, gdzie priorytetem była współrzędna x końca wierzchołka. W dalszej części tego zadania dodatkowo dodałem funkcjonalność eliminującą duplikaty.
- Struktura zdarzeń została zaimplementowana jako zrównoważone drzewo poszukiwań binarnych, gdzie wartością indeksującą była współrzędna y punktu przecięcia odcinka z miotłą.

7)

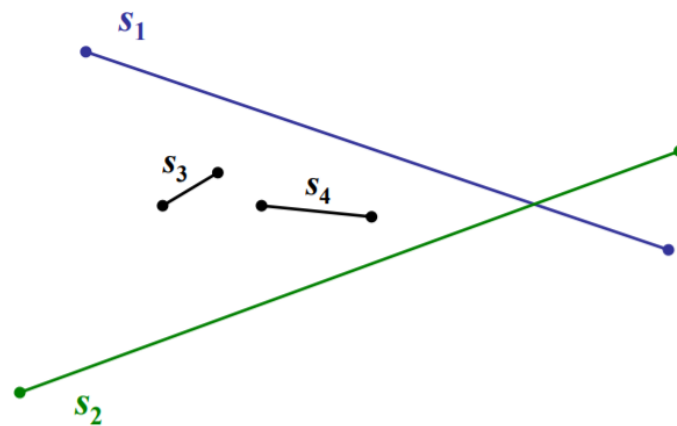
Czy konieczne były zmiany w strukturze zdarzeń?

- Zmiany w strukturze zdarzeń były potrzebne. Ponieważ tym razem chcieliśmy znaleźć wszystkie przecięcia, to w przypadku natrafienia na taki punkt dodawaliśmy go do struktury zdarzeń. Ponieważ niektóre punkty przecięcia w specyficznych sytuacjach mogły zostać wykryte wielokrotnie, musieliśmy uniknąć wystąpienia duplikatów w strukturze zdarzeń. Do kolejki priorytetowej dodana została funkcjonalność, która przed dodaniem sprawdza czy aby ten sam punkt nie pojawił się już wcześniej w kolejce (w tym wypadku nie dodaje go ponownie).

Czy w przypadku obu algorytmów konieczne były te same struktury zdarzeń?

- Nie, w przypadku znajdowania jednego punktu można było użyć zwykłej tablicy. Jej posortowanie ma złożoność  $O(n \cdot \log(n))$ , czyli taką samą jaką uzyskujemy za pomocą kolejki priorytetowej. W drugim przypadku potrzebujemy kolejki priorytetowej, ponieważ dodajemy punkty przecięcia do struktury zdarzeń, modyfikując ją.

9)



Punkt przecięcia  $s_1$  i  $s_2$  wykrywany trzy razy:

- gdy dodajemy lewy koniec  $s_1$
- gdy usuwamy prawy koniec  $s_3$
- gdy usuwamy prawy koniec  $s_4$

Poprzez sprawdzenie czy dany punkt już znajduje się w kolejce przed jego dodaniem, punkt przecięcia  $s_1$  i  $s_2$  dodajemy do struktury zdarzeń tylko gdy dodajemy lewy koniec  $s_1$ .