Obliczanie grafu widoczności

Przygotowali:

Szymon Twardosz

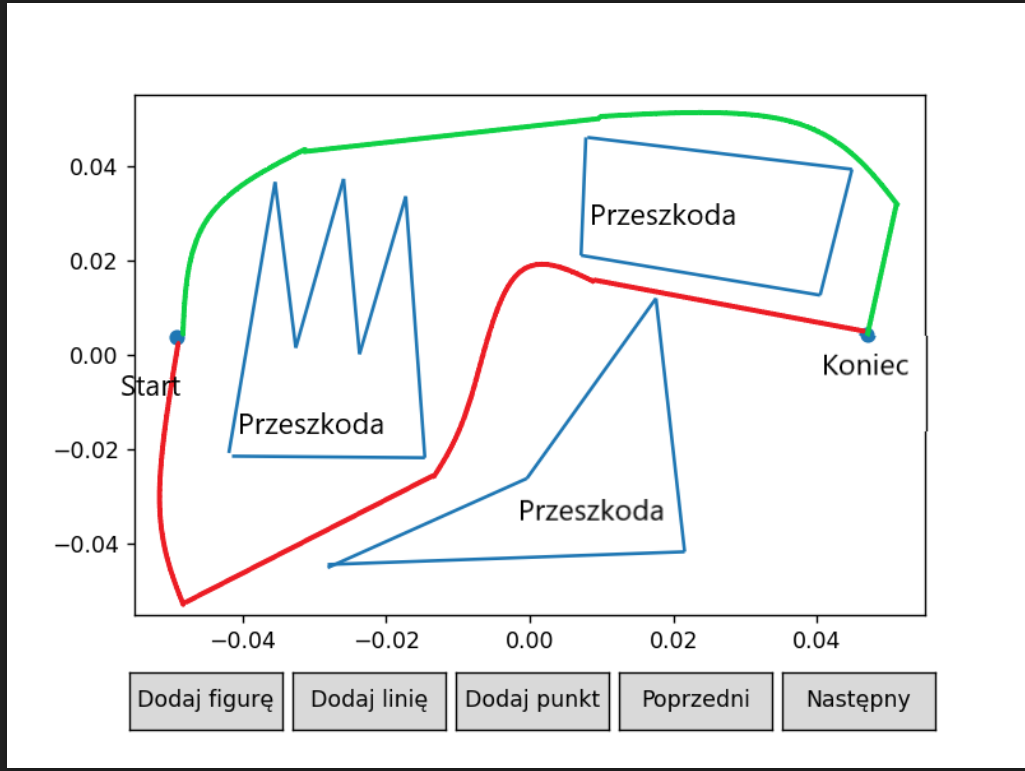
Maciej Pięta

Przedstawienie problemu

Problem polega na znalezieniu najkrótszej drogi pomiędzy punktem A a punktem B. Nie jest to jednak zwykła odległość w metryce euklidesowej. Na płaszczyźnie bowiem znajdują się przeszkody, których nasza droga nie może przecinać.

Problem ten wygląda więc następująco:

- Dane: Współrzędne punktu startowego, punktu końcowego oraz figur (przeszkód)
- Szukane: Najkrótsza ścieżka nie przecinająca żadnej przeszkody

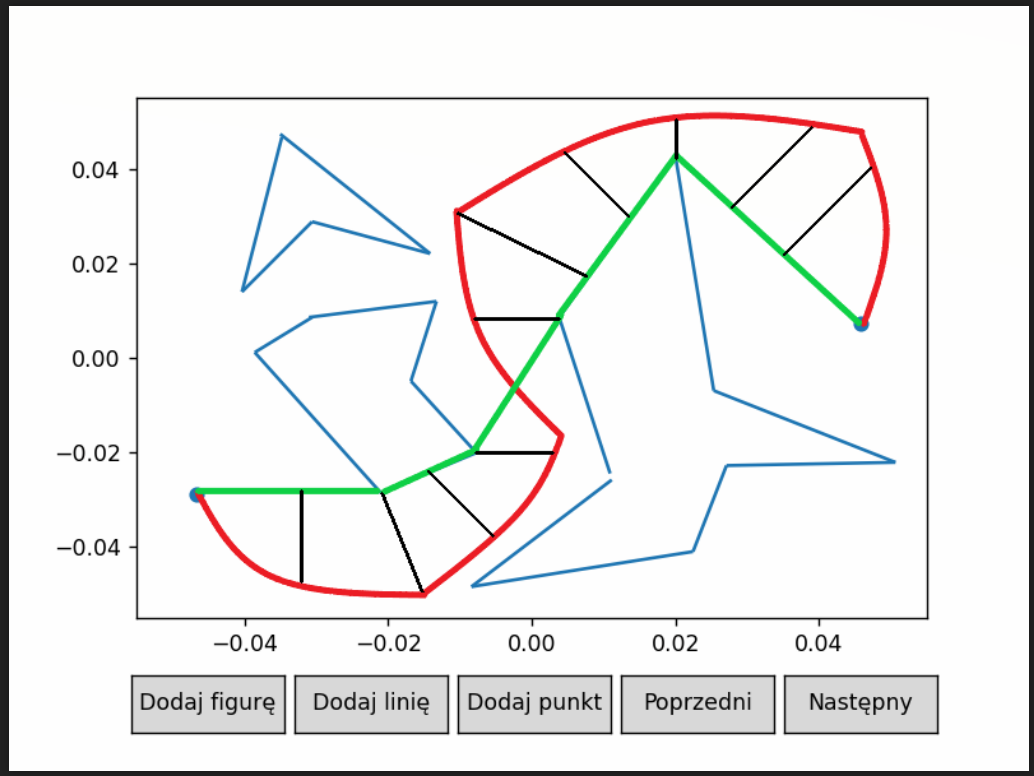


Dodatkowe założenia

Dodatkowo warto zaznaczyć, że dopuszczalne jest aby robot ,,dotykał" krawędzi przeszkód. Zakładamy, że przeszkody są więc zbiorami otwartymi.

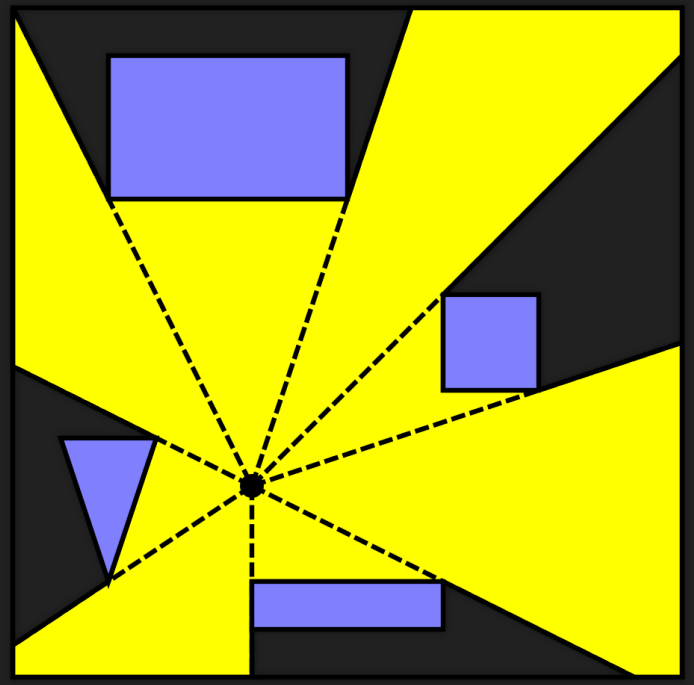
Jak znaleźć najkrótszą ścieżkę?

Obserwacja I (Lemat 1): Każda najkrótsza ścieżka pomiędzy punktem startowym a punktem końcowym to ścieżka, której wierzchołki należą do wielokątów (które są przeszkodami)



Graf widoczności

Graf widoczności – graf, złożony z wierzchołków, które ,,widzą się" wzajemnie. Krawędzie to odcinki łączące te wierzchołki, które są dla siebie widoczne. Ich waga to odległość euklidesowa.



Tworzenie grafu widoczności

Aby stworzyć graf widoczności należy dla każdego wierzchołka, znaleźć te punkty, które są dla niego widoczne (linia łącząca te dwa obiekty nie przecina żadnej przeszkody).

Algorytm:

Wejście: Zbiór S – zbiór przeszkód, punkt startowy oraz punkt końcowy

Wyjście: Graf widoczności

1. Inicjalizacja Grafu G = (V, E), gdzie V zbiór wierzchołków, E – zbiór krawędzi
2. Dla każdego wierzchołka:

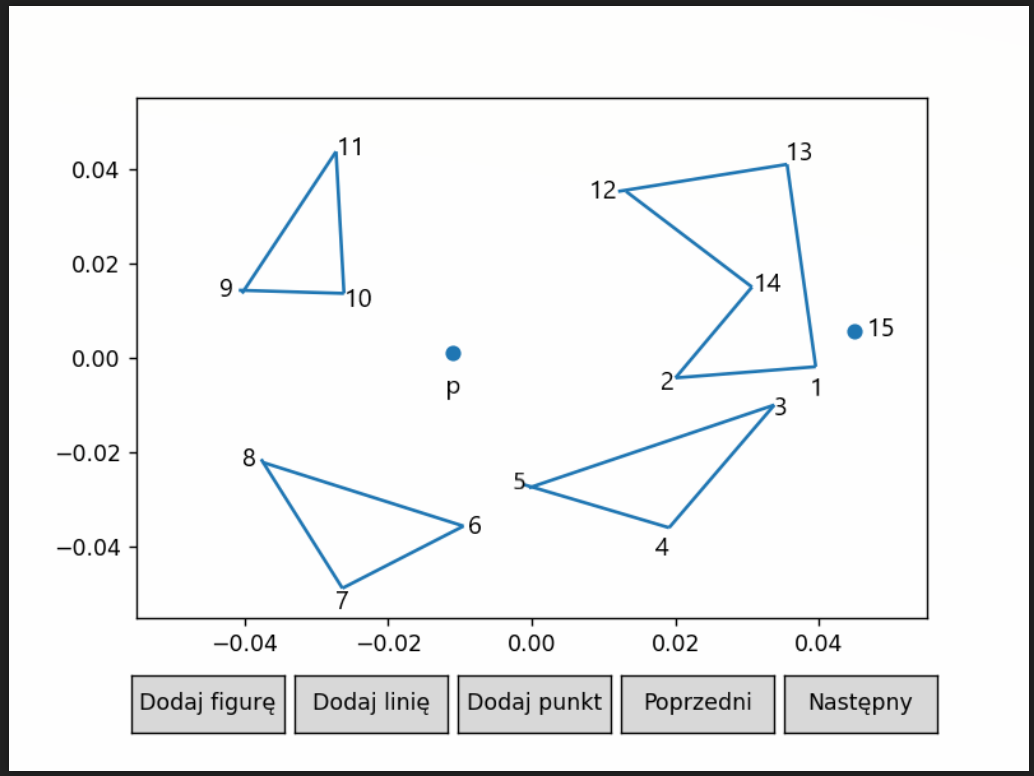
   Znajdź punkty, które są od niego widoczne oraz dodaj krawędzie łączące te punkty z rozważanym wierzchołkiem do grafu
3. Zwróć utworzony graf

Szukanie widocznych wierzchołków

Do znajdywania widocznych wierzchołków wygodne będzie wykorzystanie **miotły**.

Struktura zdarzeń: Wierzchołki ułożone według kąta jaki tworzy odcinek zakończony w danym wierzchołku (wychodzący z aktualnie rozważanego punktu p) a osią OX

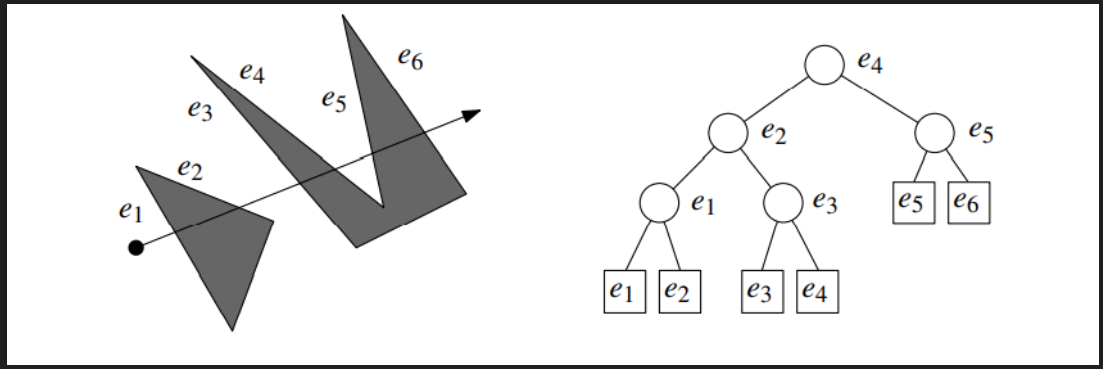
Struktura stanu: Drzewo binarne przechowujące aktualnie przecięte odcinki



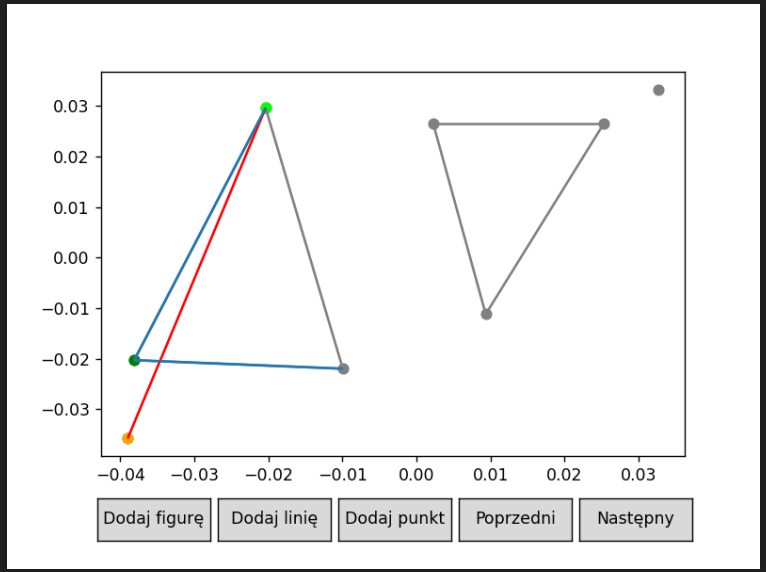
Struktura stanu miotły

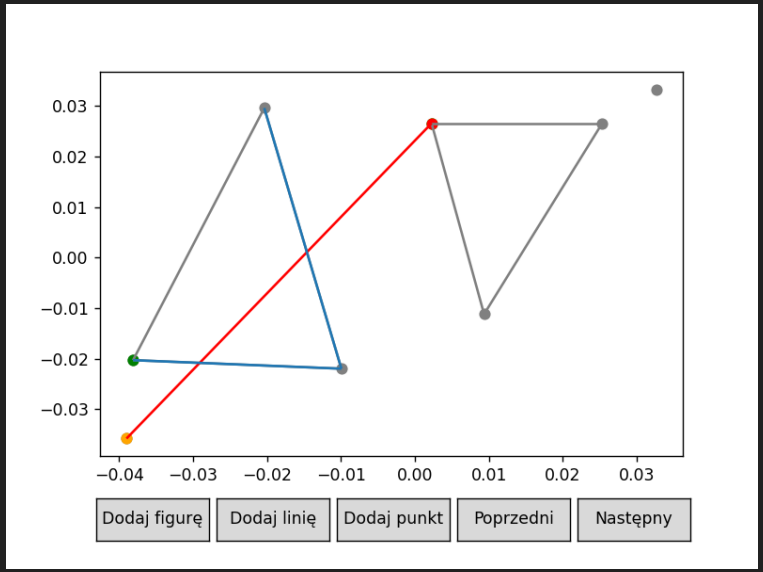
Oznaczenia: ρ – miotła, p – punkt miotły początkowy, w – punkt końcowy miotły

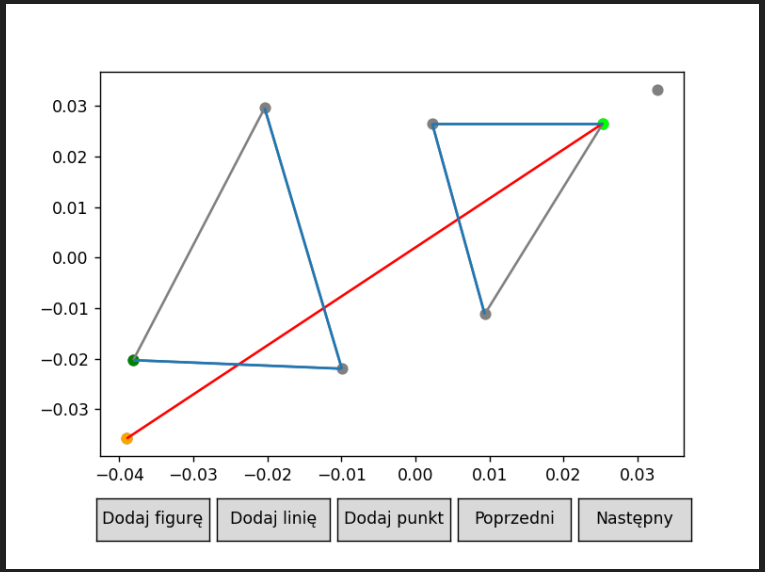
W strukturze stanu znajdują się odcinki aktualnie przecięte przez miotłę. Ich kolejność nie jest przypadkowa. Wcześniej znajdują się linie znajdujące się bliżej (metryka euklidesowa) punktu startowego p.



Struktura stanu miotły







Szukanie widocznych wierzchołków - Algorytm

Algorytm:

Wejście: S – zbiór przeszkód oraz punkt początkowy oraz startowy, p – punkt dla którego szukamy widocznych wierzchołków

Wyjście: W – zbiór widocznych wierzchołków

1. Posortuj wierzchołki według kąta jaki tworzy prosta od puntu p do wierzchołka a oś OX
2. Znajdź wszystkie linie które przecinają prostą równoległa do OX wychodzącą z punktu p
3. Dla każdego wierzchołka (w):

   Jeżeli jest widoczny to: Dodaj go do zbioru W

   Dodaj do struktury stanu miotły krawędzie wychodzące z w leżące po prawej stronie miotły.

   Usuń z struktury stanu miotły krawędzie wychodzące z w leżące po lewej stronie miotły
4. Zwróć W

Czy wierzchołek jest widoczny? - Algorytm

Algorytm:

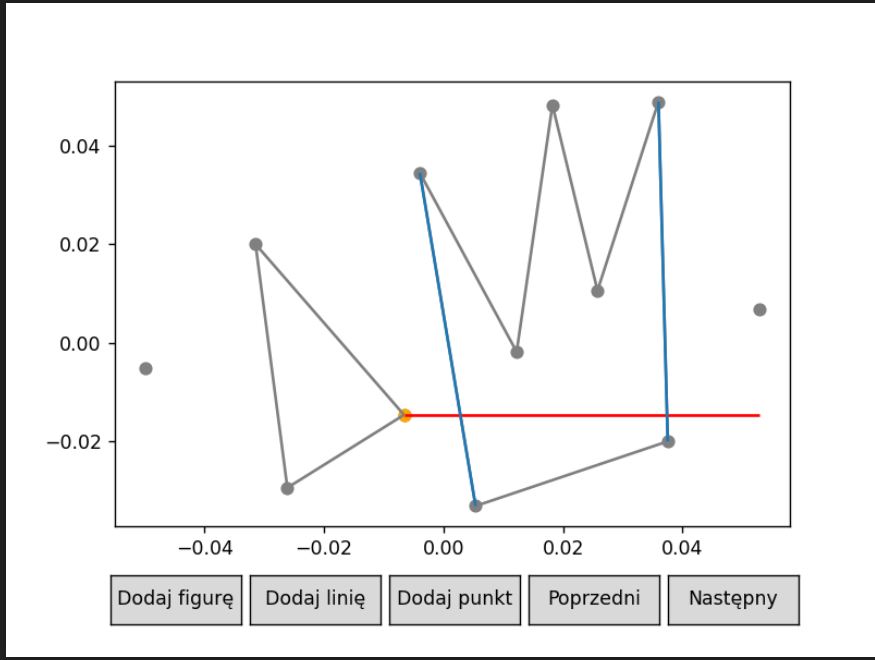
Wejście: p –wierzchołek dla którego szukamy widocznych punktów, w_i – wierzchołek dla którego sprawdzamy czy jest widoczny, T – struktura stanu miotły

Wyjście: Wartość logiczna (jest widoczny bądź nie)

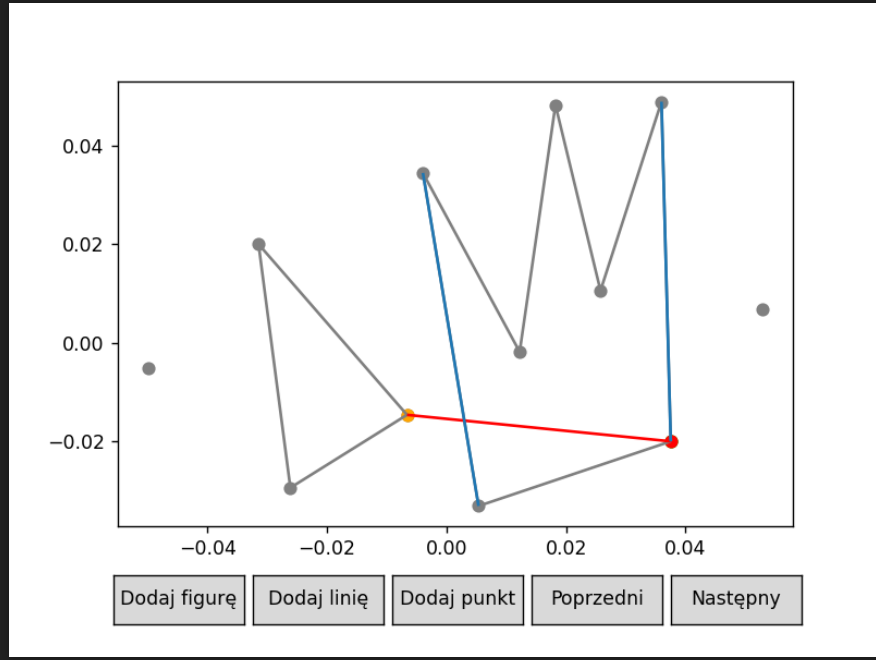
1. Jeżeli pw_i przecina wnętrze przeszkody, której w jest wierzchołkiem: Zwróć Fałsz
2. Jeżeli w_i jest pierwszym rozważanym wierzchołkiem lub w_{i-1} nie znajduje się na pw_i to:
   Znajdź w T odcinek o najmniejszej wadze (znajdujący się najbliżej p)
   Jeżeli taki istnieje oraz przecina pw_i: Zwróć Fałsz
   W przeciwnym przypadku zwróć Prawdę
3. Jeżeli w_{i-1} jest nie widoczne (dla p): Zwróć Fałsz
4. W przeciwnym przypadku: Znajdź w T odcinek który przecina $w_{i-1}w_i$
   Jeżeli odcinek istnieje: Zwróć Fałsz
   W przeciwnym przypadku: Zwróć Prawdę

Przykład działania algorytmu

Krok 1 – Miotła równoległa do osi OX

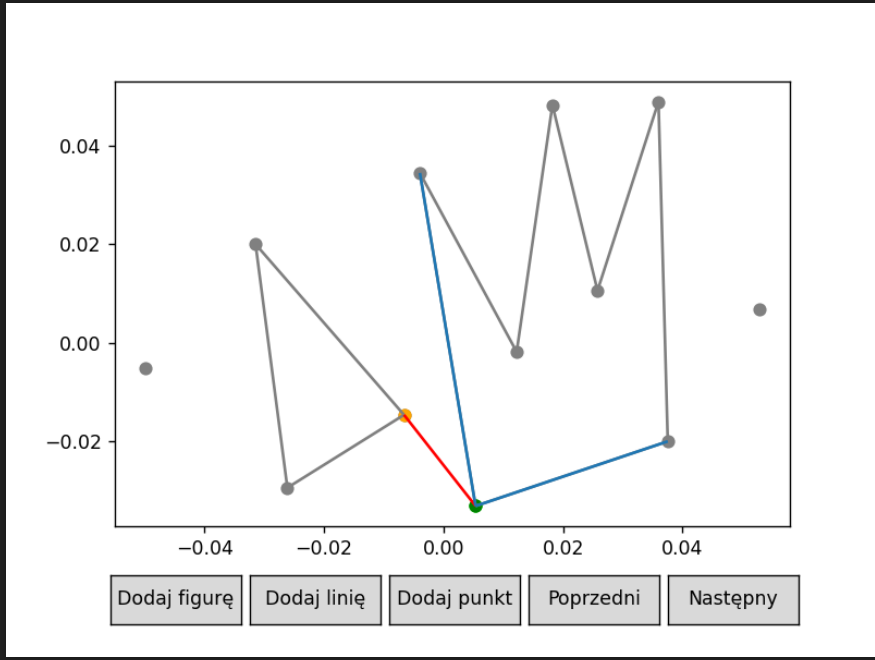


Krok 2 – Punkt nie widoczny

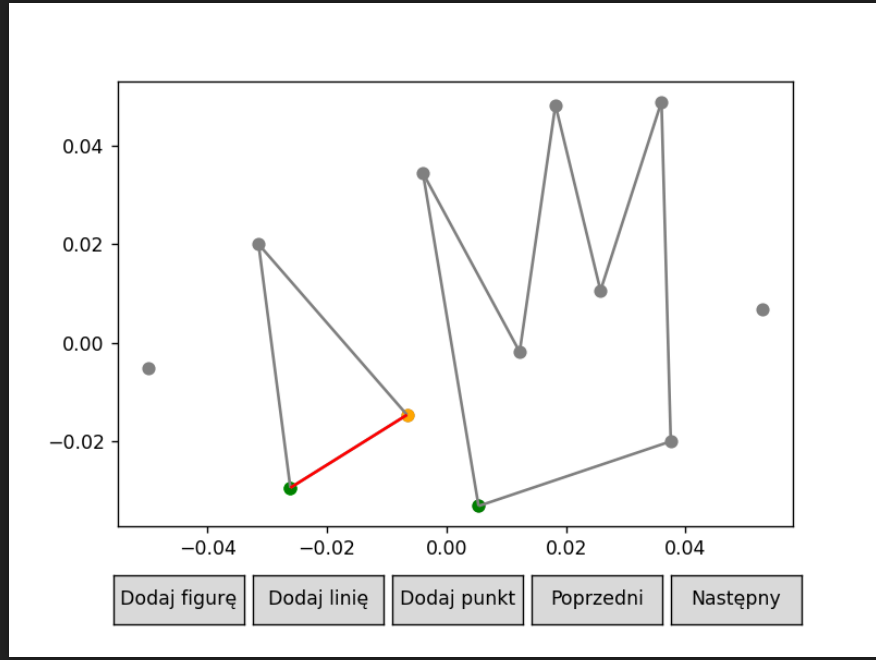


Przykład działania algorytmu

Krok 3 – Punkt widoczny

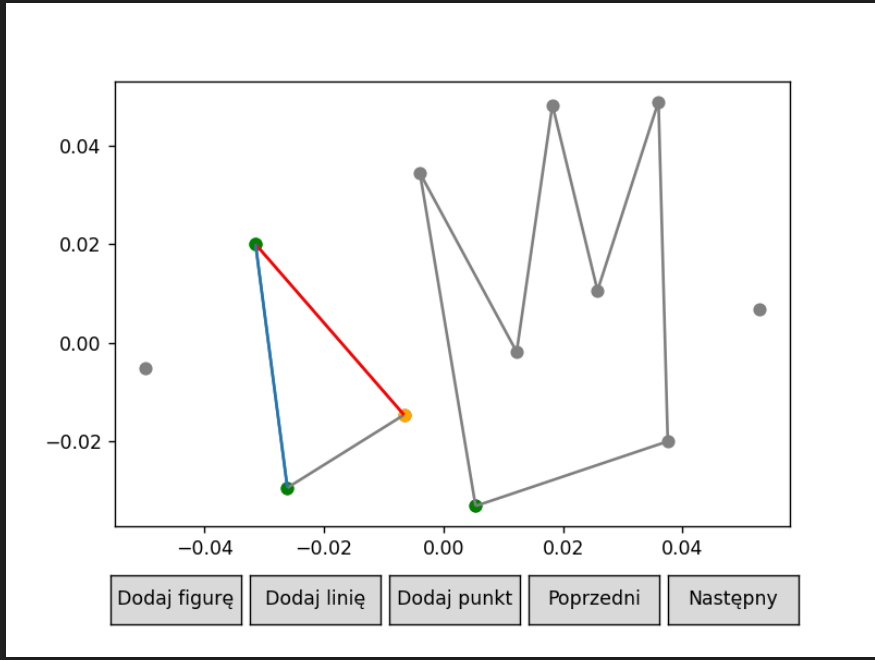


Krok 4 – Punkt widoczny

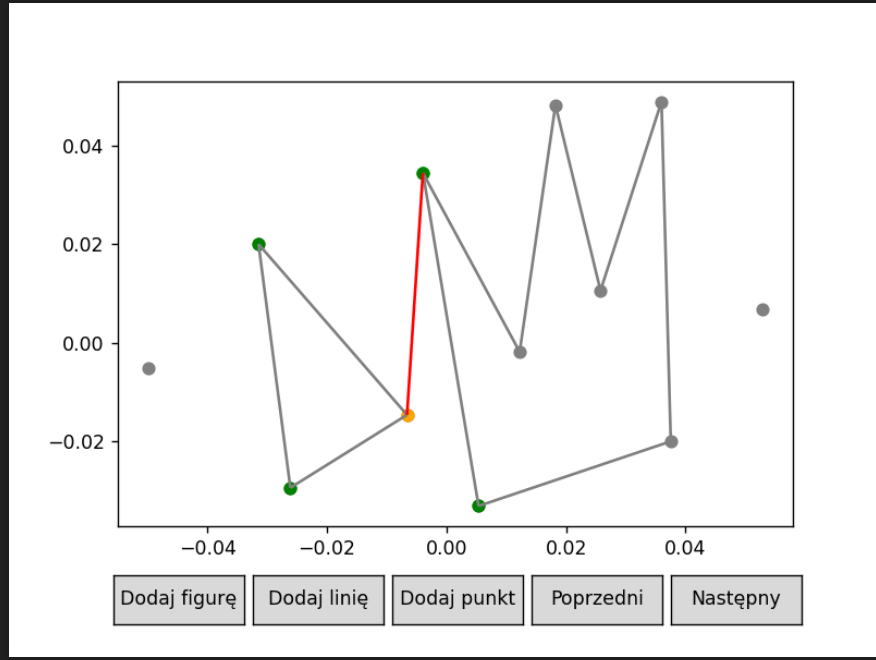


Przykład działania algorytmu

Krok 5 – Punkt widoczny

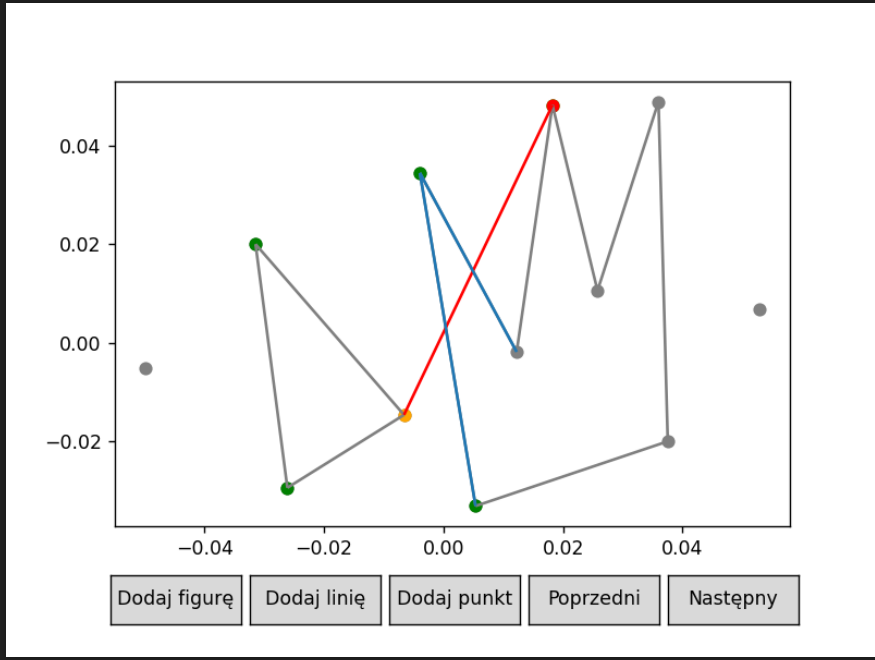


Krok 6 – Punkt widoczny

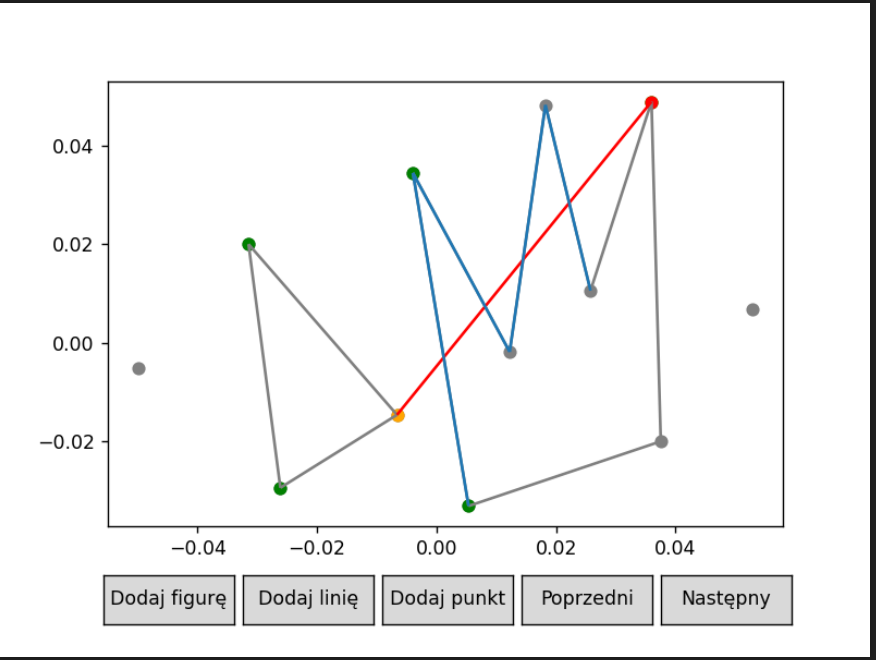


Przykład działania algorytmu

Krok 7 – Punkt nie widoczny



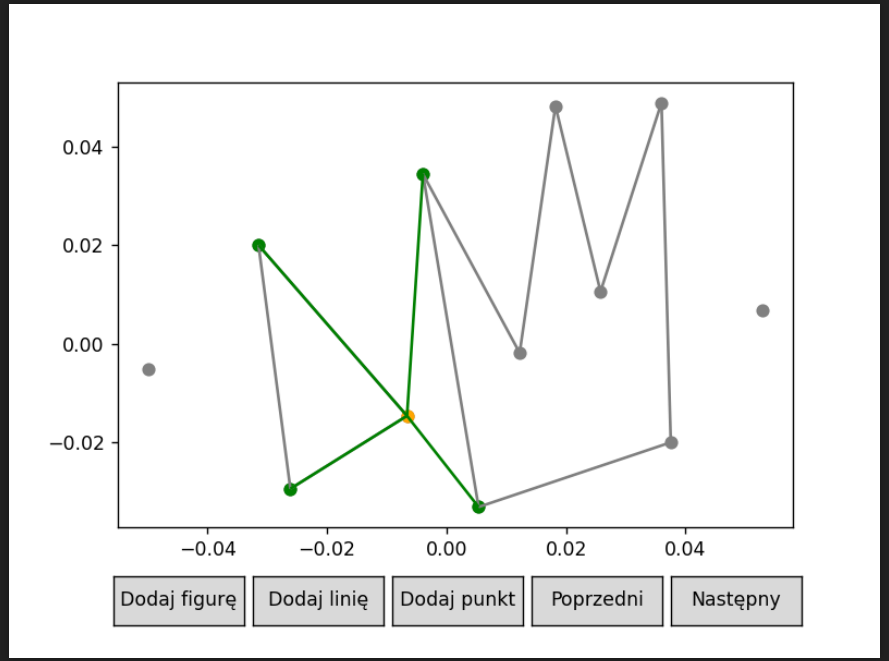
Krok 8 – Punkt nie widoczny



Przykład działania algorytmu

Trzy kroki zostały pominięte

Krok 12 – Dodane wierzchołki



Szukanie najkrótszej ścieżki

Algorytm Dijkstry – algorytm służący do znajdowania najkrótszej ścieżki z pojedynczego źródła w grafie o nieujemnych wagach.

Algorytm:

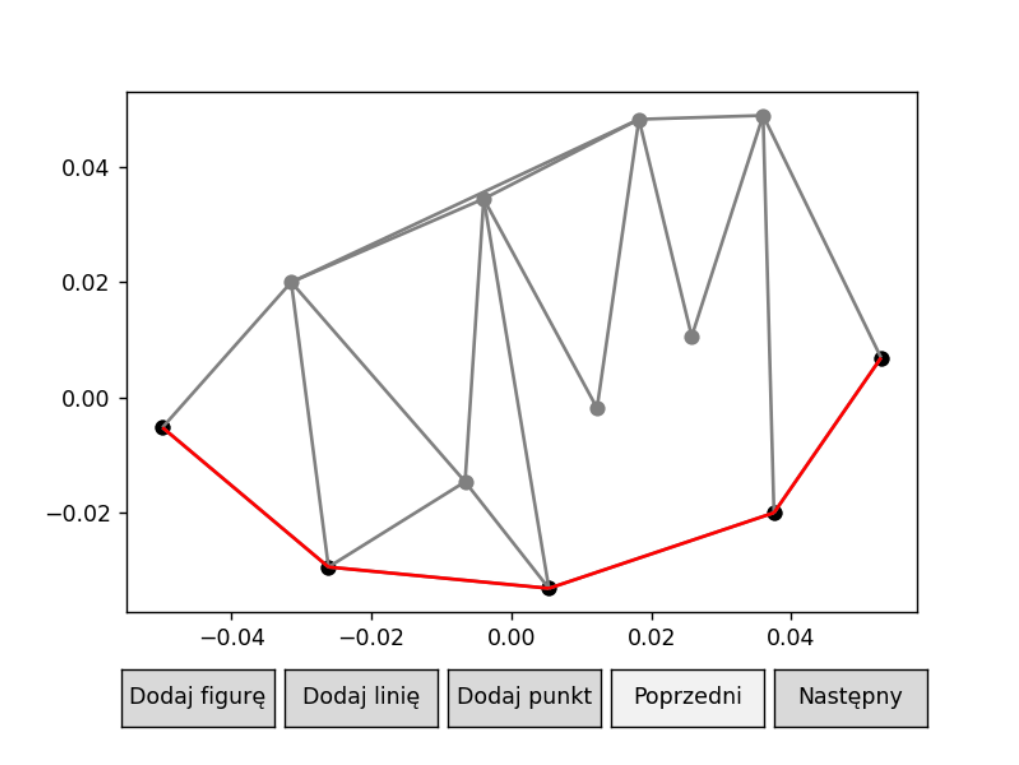
Wejście: Graf G = (V, E), wierzchołki: startowy s, końcowy t

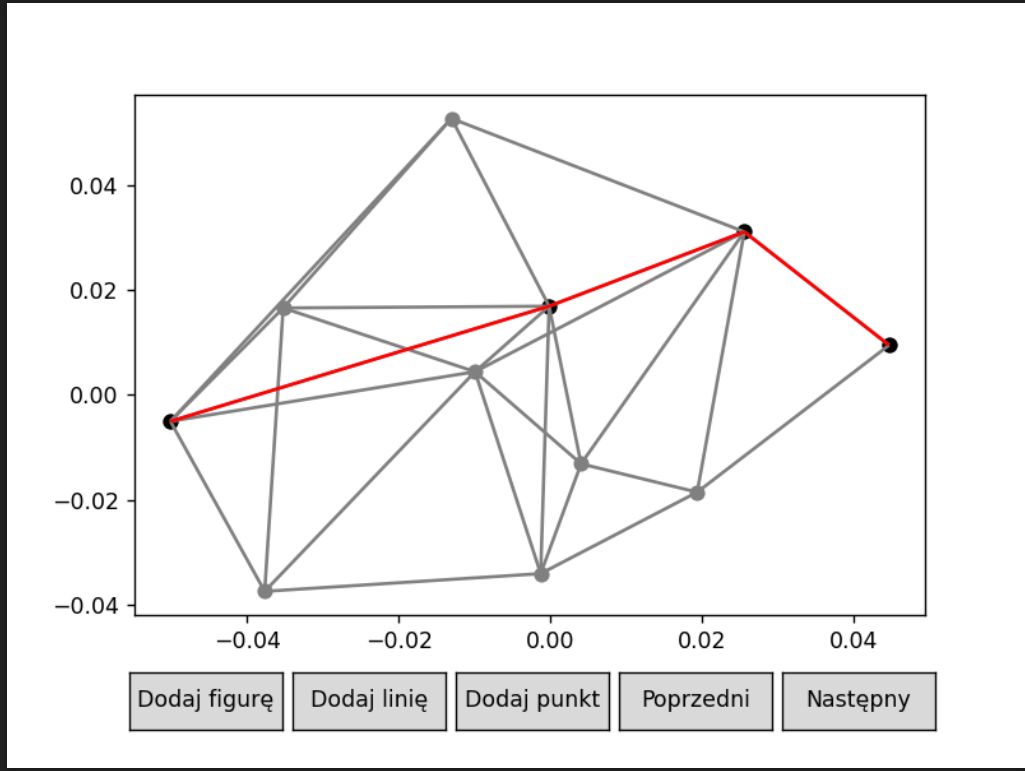
1. Stwórz kolejkę priorytetową
2. Stwórz listę aktualnych odległości od źródła. Ustaw wszystkie wartości na ∞
3. Dodaj do kolejki wierzchołek s z wagą 0. Taką samą wartość ustaw w tablicy odległości dla wierzchołka t
4. Dopóki kolejka nie jest pusta:

   Wyjmij wierzchołek v z kolejki o minimalnym oszacowaniu odległości.

   Dla każdej krawędzi {u,v} (u to sąsiad v) wykonaj relaksację (ponowne oszacowanie odległości).

Przykładowe rozwiązania





Złożoność obliczeniowa

Złożoność obliczeniowa poszczególnych algorytmów:

- Algorytm tworzenia grafu widoczności – $O(n^2 \log n)$, gdzie n – liczba krawędzi wszystkich przeszkód
- Algorytm Dijkstry – $O(n \log n + k)$, gdzie k – liczba krawędzi utworzonego grafu

Podsumowując, złożoność obliczeniowa problemu grafu widoczności wynosi $O(n^2 \log n)$

Literatura

- Thomas H. Corner, Charles E. Leiserton, Ronald L. Rivest, ,,Wprowadzenie do algorytmów”
- Mark de Berg „Geometria obliczeniowa – Algorytmy i Zastosowania”
- Wikipedia – obraz slajd 5

Dziękujemy za uwagę