

# Problem Komiwożera

## Travelling Salesman Problem

# Zadanie

- Dany jest zbiór miast

# Zadanie

- Dany jest zbiór miast

## **Przykład**

Dane jest 7 miast:

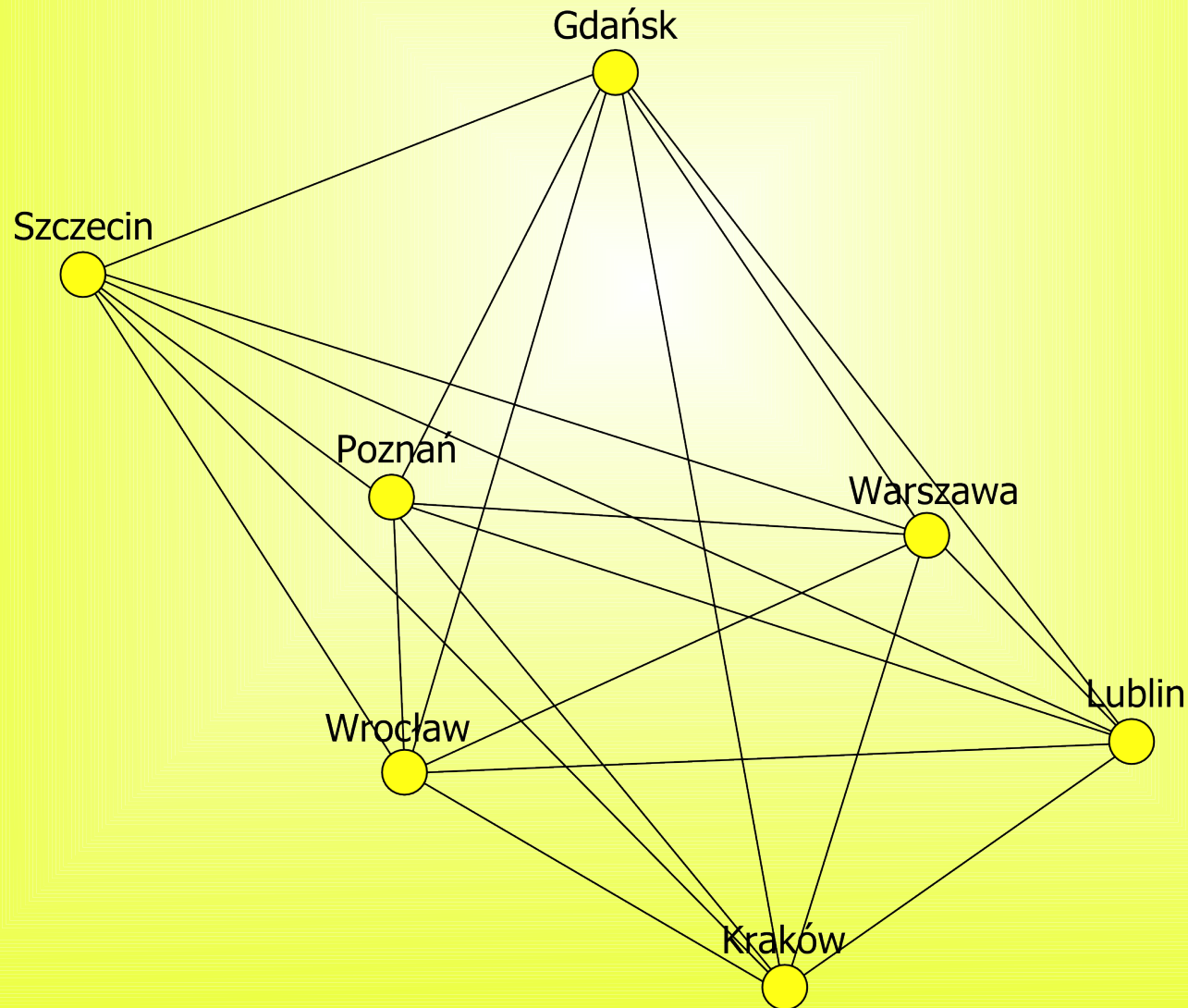
- Wrocław,
- Szczecin,
- Gdańsk,
- Lublin,
- Warszawa,
- Poznań,
- Kraków.

# Zadanie

- Wszystkie miasta połączone są ze sobą

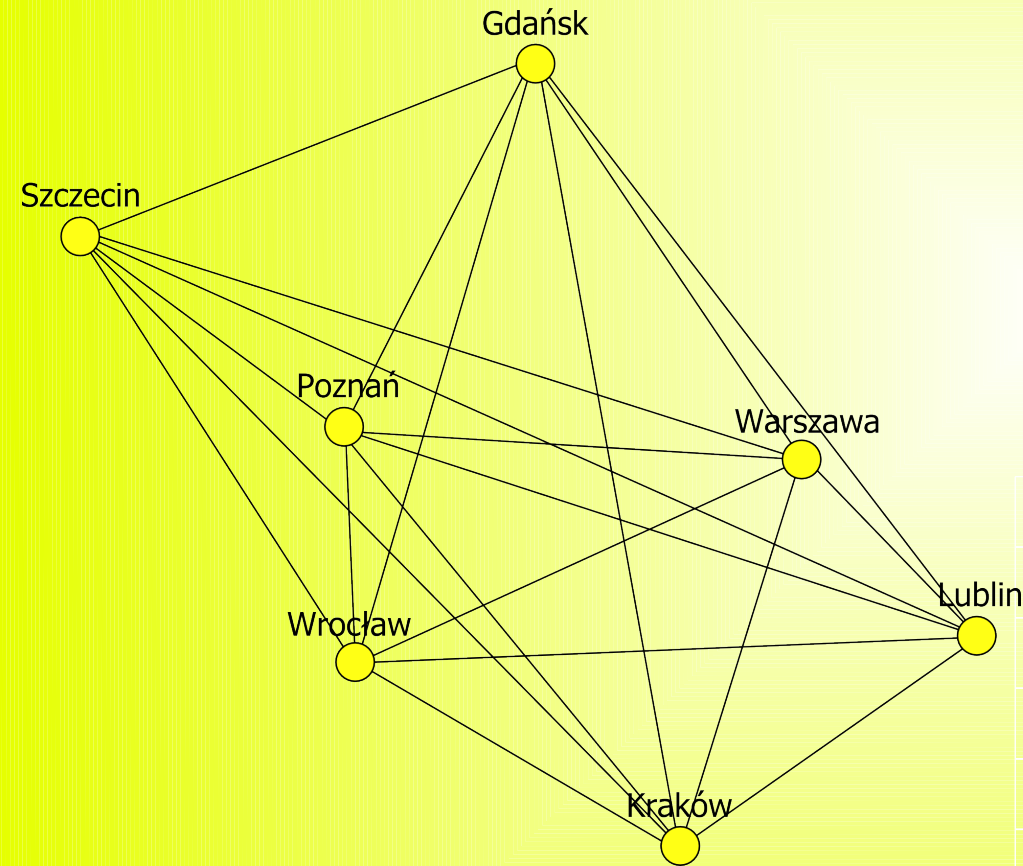
# Zadanie

- Wszystkie miasta połączone są ze sobą



# Zadanie

- Dane są odległości między wszystkimi miastami



	SZC	WRO	POZ	GDA	WAW	LUB	KRK
SZC	0	364	235	366	565	690	662
WRO	364	0	169	435	342	431	272
POZ	235	169	0	296	319	459	424
GDA	366	435	296	0	348	517	603
WAW	565	342	319	348	0	168	293
LUB	690	431	459	517	168	0	270
KRK	662	272	424	603	293	270	0

# Zadanie

- Zadaniem jest obliczenie najkrótszej trasy komiwojażera, który ma odwiedzić wszystkie miasta i wrócić doń z powrotem.

# Zadanie

- Zadaniem jest obliczenie najkrótszej trasy komiwojażera, który ma odwiedzić wszystkie miasta i wrócić doń z powrotem.

## **Przykład**

Kurier wyrusza z Wrocławia i ma odwiedzić każde z pozostałych sześciu miast dokładnie raz po czym wrócić do Wrocławia. Trasa ma być najkrótsza z możliwych.



# Rodzaje TSP

- Symetryczny Problem Komiwojażera (STSP),
- Asymetryczny Problem Komiwojażera (ATSP),
- Oba te problemy są NP-trudne.

# Algorytm NEH

- Algorytm NEH umożliwia wyznaczenie krótkiej (nie zawsze najkrótszej) trasy dla TSP
  - wybieramy wierzchołek startowy i umieszczamy go na liście dwukrotnie,
  - wstawiamy kolejne miasto na każdą z pozycji w liście (oprócz pozycji pierwszej i ostatniej) i sprawdzamy, który z wyników jest najmniejszy; na tej pozycji pozostawiamy miasto,
  - powtarzamy powyższy krok aż wyczerpiemy wszystkie miasta.

# Przykład działania

- Wyjeżdżamy z Wrocławia

WRO WRO  $\rightarrow$  0

# Przykład działania

- Dodajemy Szczecin do uporządkowania

WRO SZC WRO  $\rightarrow 364+364 = 728$  \*

# Przykład działania

- Dodajemy Poznań do uporządkowania

WRO POZ SZC WRO  $\rightarrow 169+235+364 = 768$  \*

WRO SZC POZ WRO  $\rightarrow 364+235+169 = 768$

# Przykład działania

- Dodajemy Gdańsk do uporządkowania

WRO GDA POZ SZC WRO  $\rightarrow 435+296+235+364 = 1330$

WRO POZ GDA SZC WRO  $\rightarrow 169+296+366+364 = 1195 *$

WRO POZ SZC GDA WRO  $\rightarrow 169+235+366+435 = 1205$

# Przykład działania

- Dodajemy Warszawę do uporządkowania

WRO WAW POZ GDA SZC WRO  $\rightarrow 342+319+296+366+364 = 1687$

WRO POZ WAW GDA SZC WRO  $\rightarrow 169+319+348+366+364 = 1566 *$

WRO POZ GDA WAW SZC WRO  $\rightarrow 169+296+348+565+364 = 1742$

WRO POZ GDA SZC WAW WRO  $\rightarrow 169+296+366+565+342 = 1738$

# Przykład działania

- Dodajemy Lublin do uporządkowania

WRO LUB POZ WAW GDA SZC WRO  $\rightarrow 431+459+319+348+366+364 = 2287$

WRO POZ LUB WAW GDA SZC WRO  $\rightarrow 169+459+168+348+366+364 = 1874 *$

WRO POZ WAW LUB GDA SZC WRO  $\rightarrow 169+319+168+517+366+364 = 1903$

WRO POZ WAW GDA LUB SZC WRO  $\rightarrow 169+319+348+517+690+364 = 2407$

WRO POZ WAW GDA SZC LUB WRO  $\rightarrow 169+319+348+366+690+431 = 2323$



# Przykład działania

- Dodajemy Kraków do uporządkowania

WRO KRK POZ LUB WAW GDA SZC WRO →  $272+424+459+168+348+366+364 = 240$

WRO POZ KRK LUB WAW GDA SZC WRO →  $169+424+270+168+348+366+364 = 2109$  \*

WRO POZ LUB KRK WAW GDA SZC WRO →  $169+459+270+293+348+366+364 = 2269$

WRO POZ LUB WAW KRK GDA SZC WRO →  $169+459+168+293+603+366+364 = 2422$

WRO POZ LUB WAW GDA KRK SZC WRO →  $169+459+168+348+603+662+364 = 2773$

WRO POZ LUB WAW GDA SZC KRK WRO →  $169+459+168+348+366+662+272 = 2444$

# Przykład działania

- Odnajdujemy najlepsze\* uporządkowanie

WRO KRK POZ LUB WAW GDA SZC WRO →  $272+424+459+168+348+366+364 = 240$

**WRO POZ KRK LUB WAW GDA SZC WRO →  $169+424+270+168+348+366+364 = 2109$  \***

WRO POZ LUB KRK WAW GDA SZC WRO →  $169+459+270+293+348+366+364 = 2269$

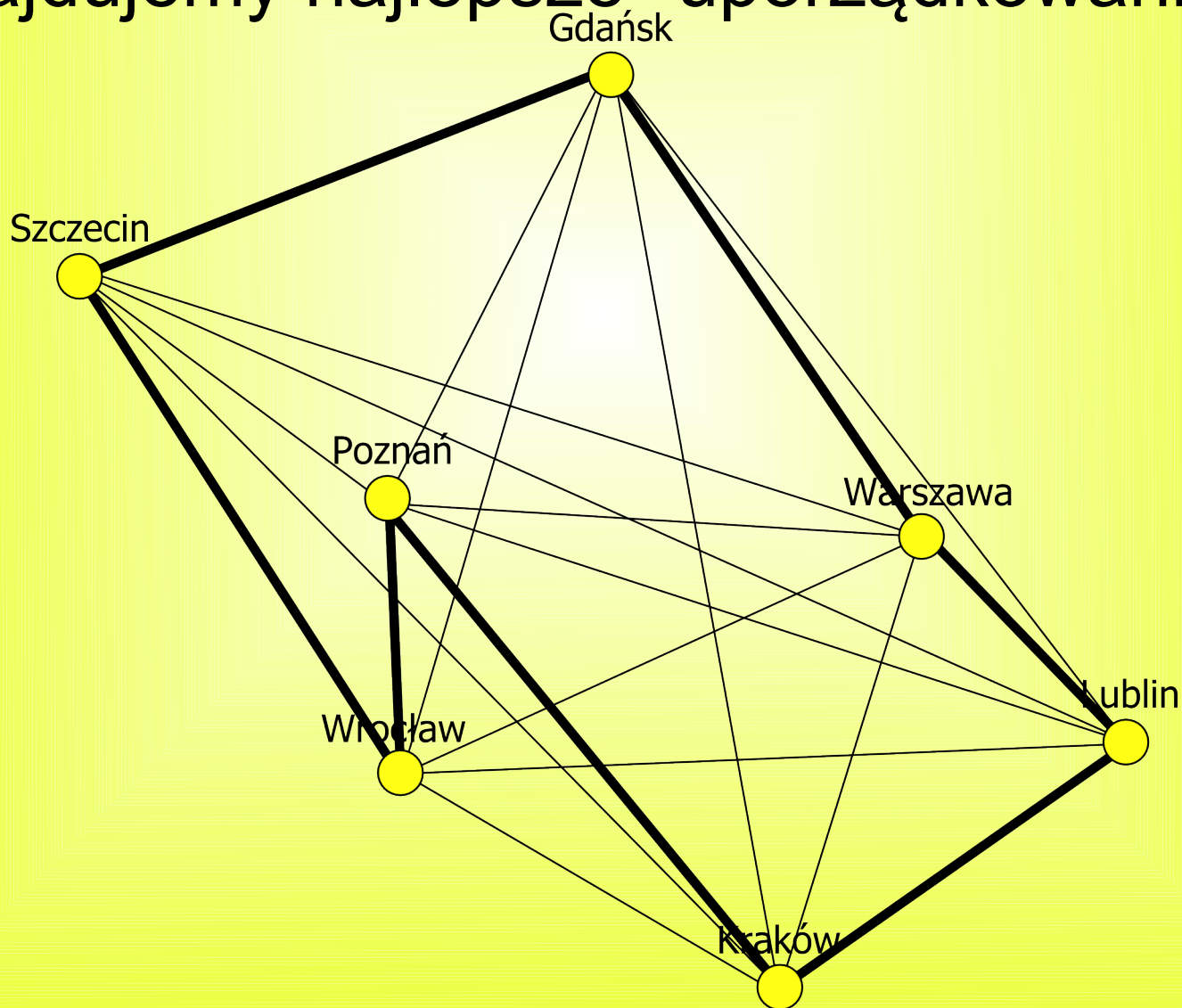
WRO POZ LUB WAW KRK GDA SZC WRO →  $169+459+168+293+603+366+364 = 2422$

WRO POZ LUB WAW GDA KRK SZC WRO →  $169+459+168+348+603+662+364 = 2773$

WRO POZ LUB WAW GDA SZC KRK WRO →  $169+459+168+348+366+662+272 = 2444$

# Przykład działania

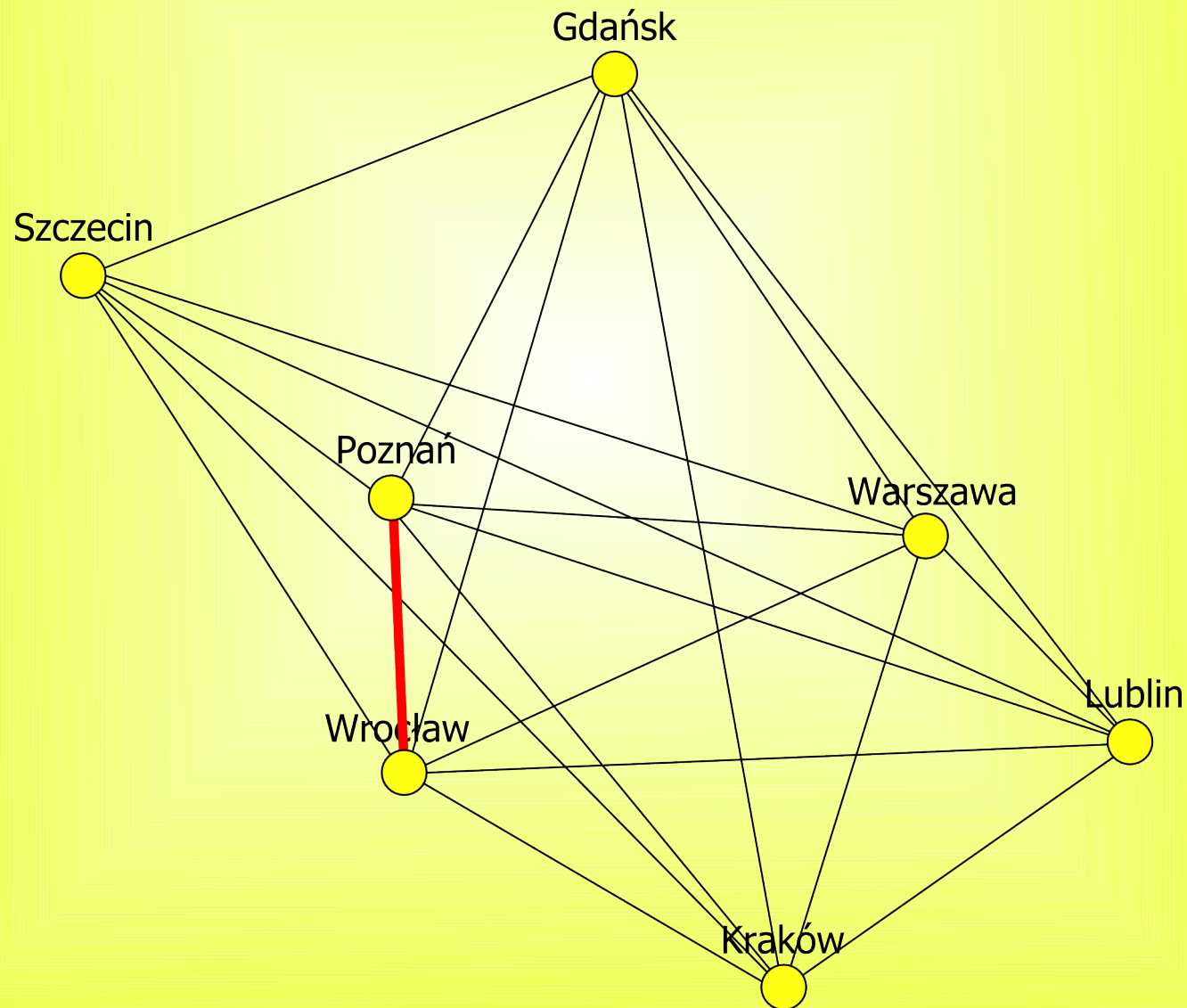
- Odnajdujemy najlepsze\* uporządkowanie



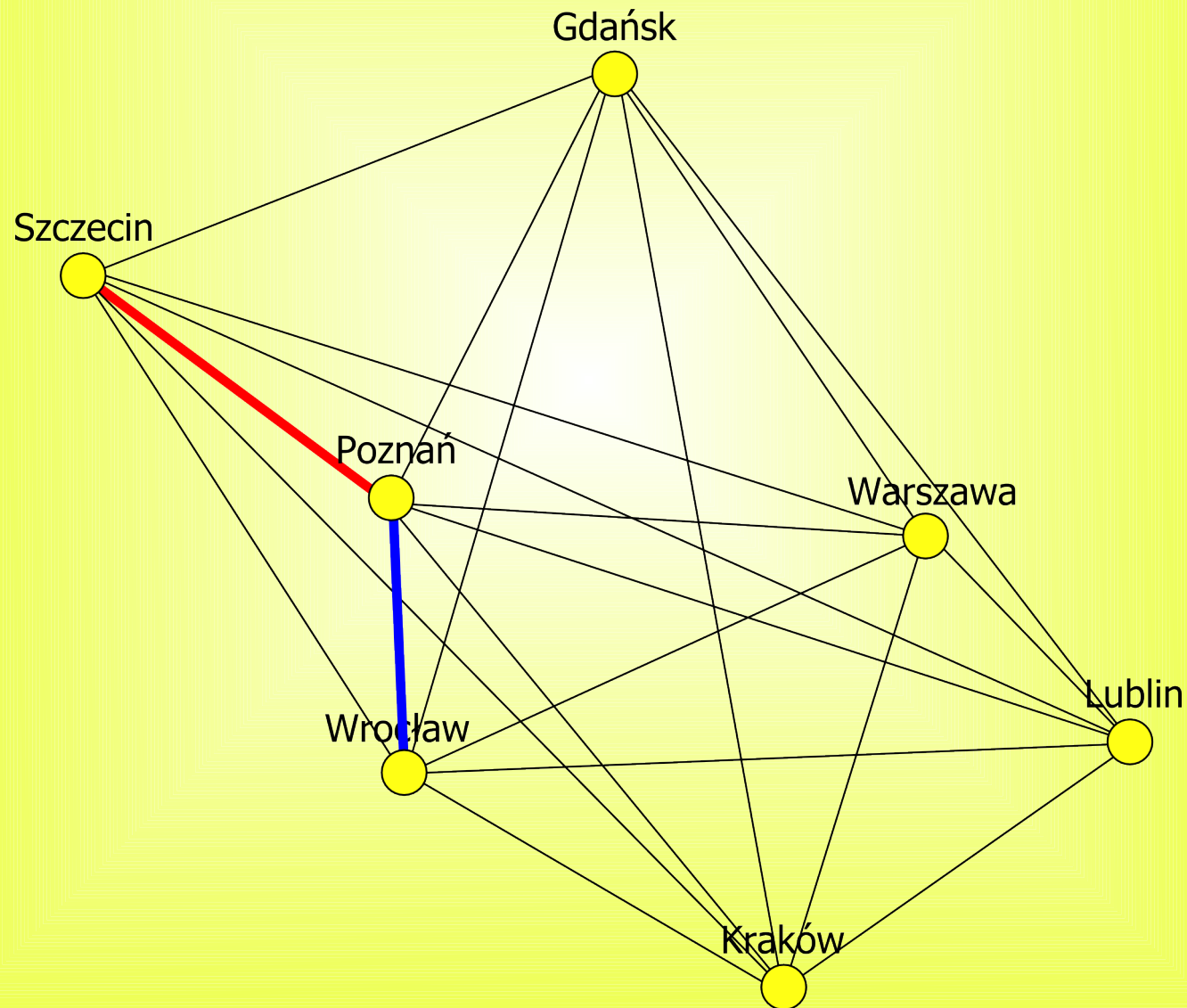
# Algorytm najkrótszej (lokalnie) ścieżki

- wybieramy miasto początkowe,
- do listy dodajemy to miasto (z jeszcze niedodanych), które jest najbliżej ostatniego dodanego do listy,
- jeśli lista miast do wyboru jest pusta – dodajemy na koniec miasto początkowe; jeśli nie – wracamy do punktu drugiego,
- koniec algorytmu

# Algorytm najkrótszej (lokalnie) ścieżki

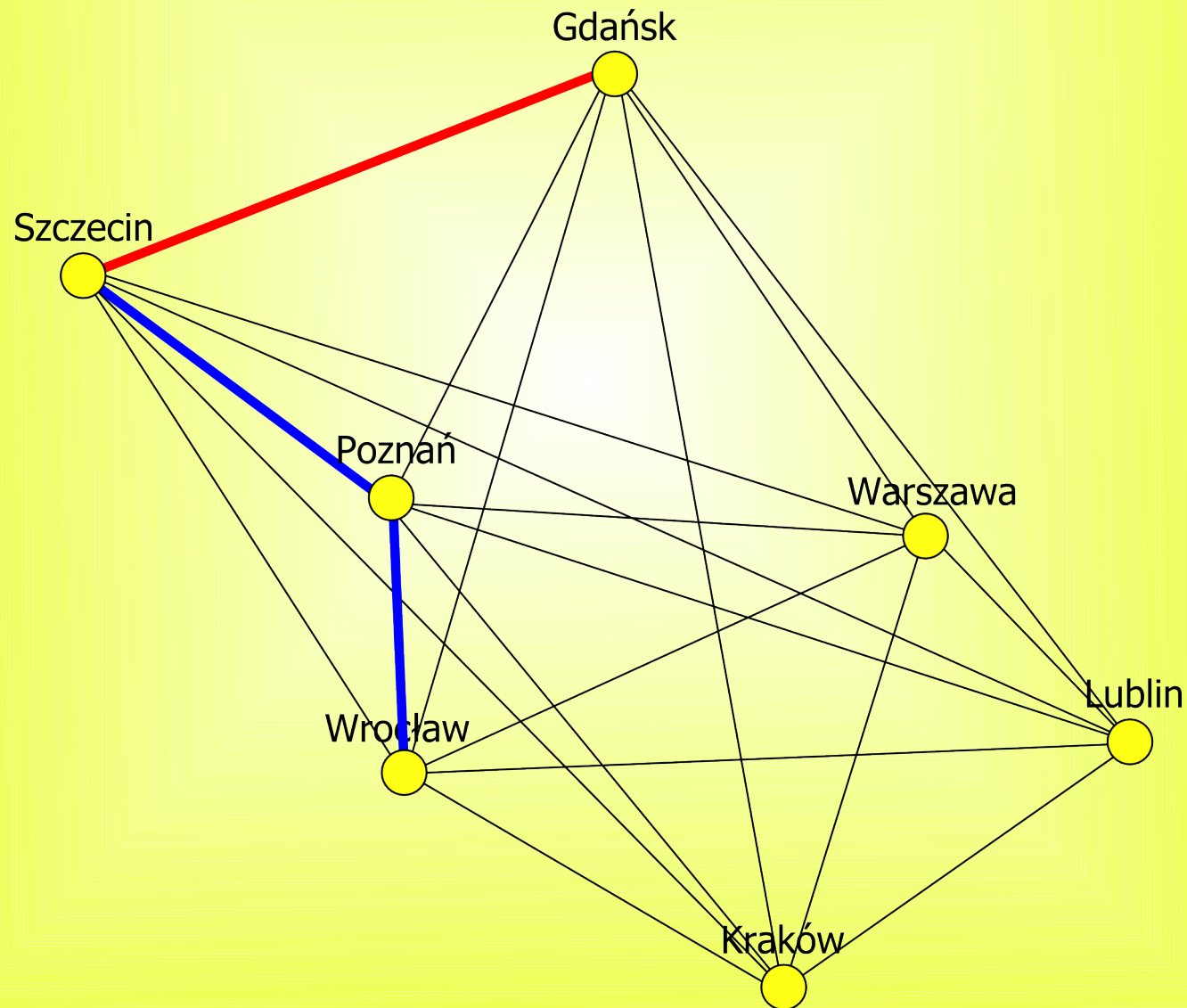


# Algorytm najkrótszej (lokalnie) ścieżki

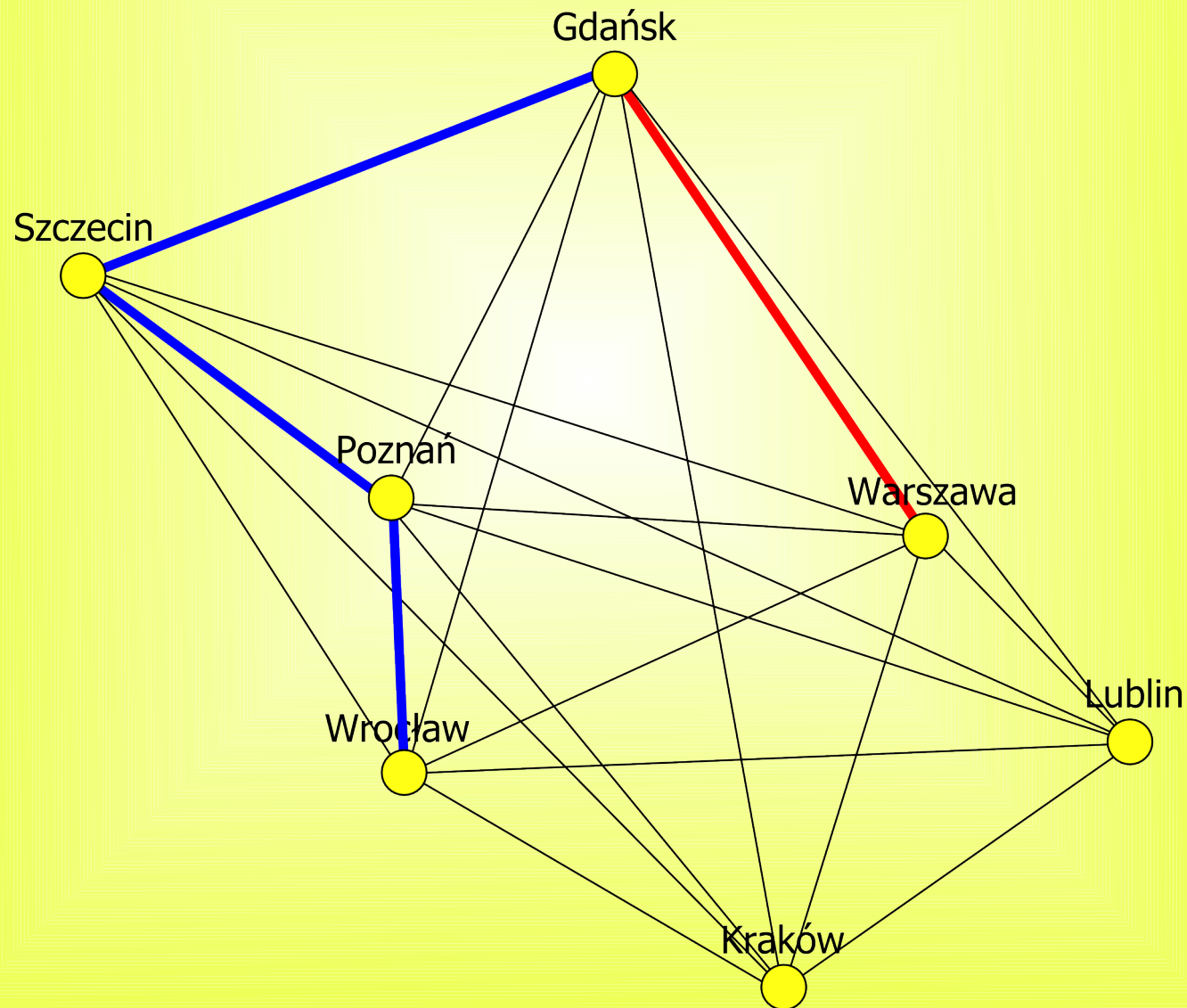




# Algorytm najkrótszej (lokalnie) ścieżki

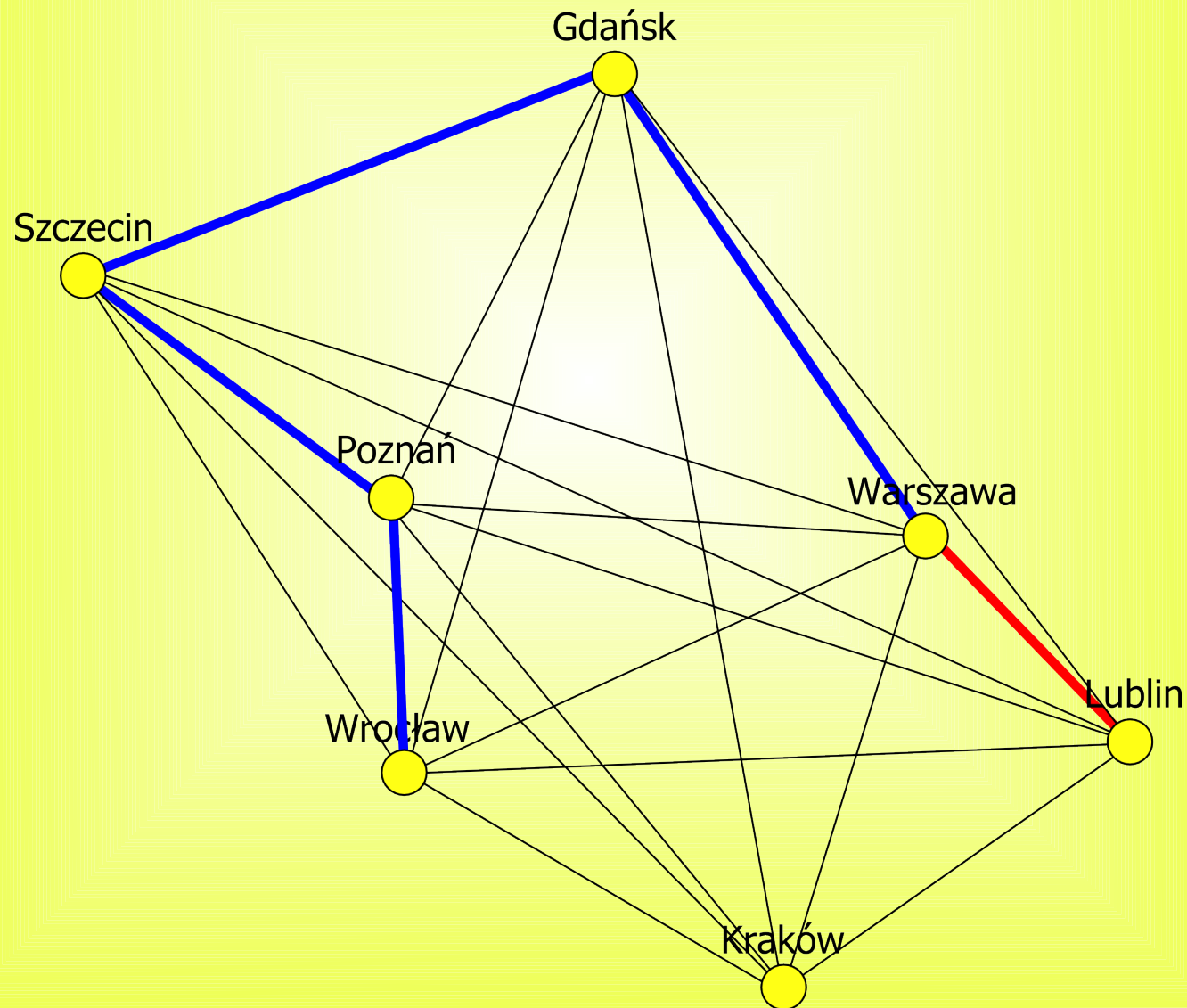


# Algorytm najkrótszej (lokalnie) ścieżki

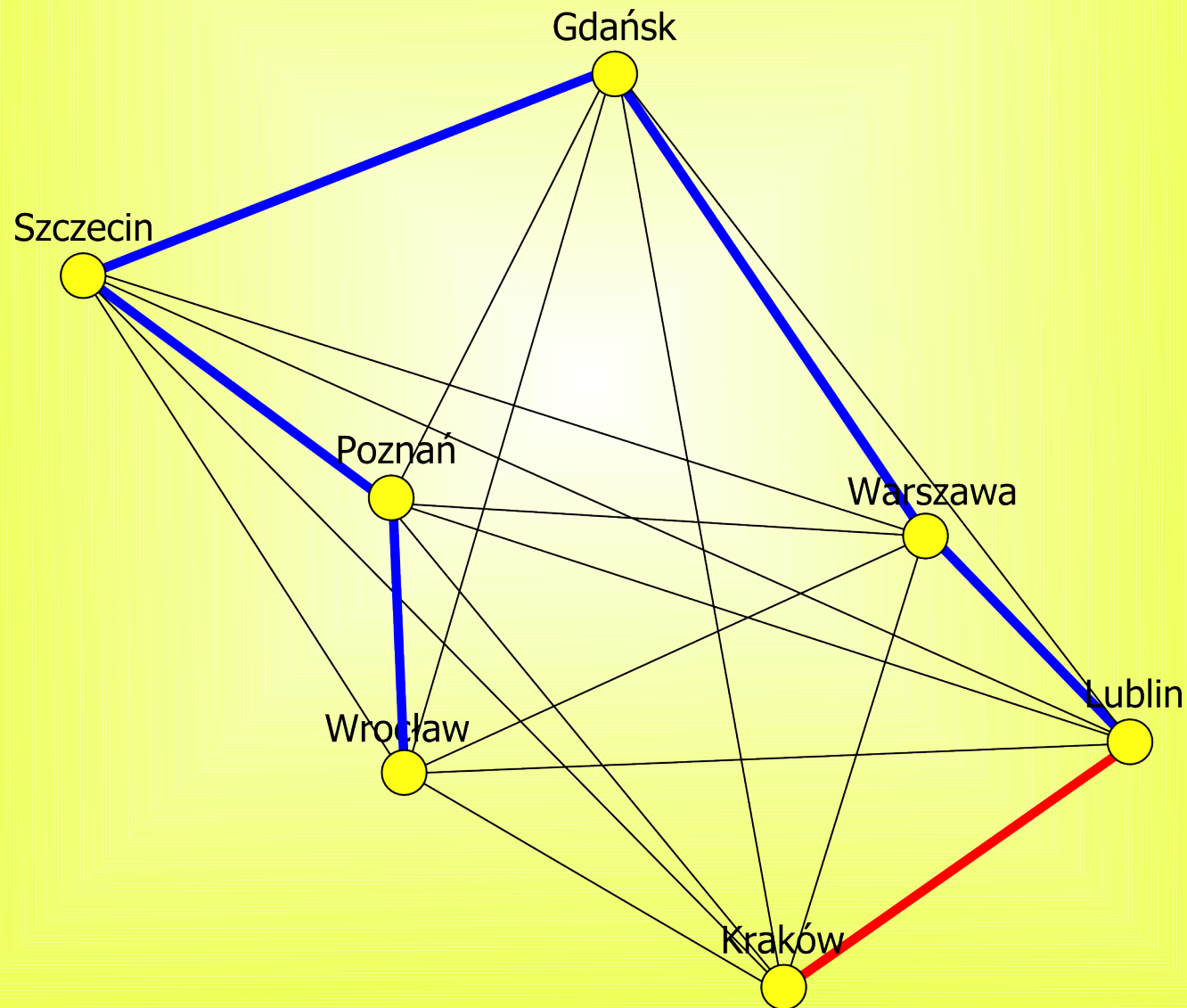




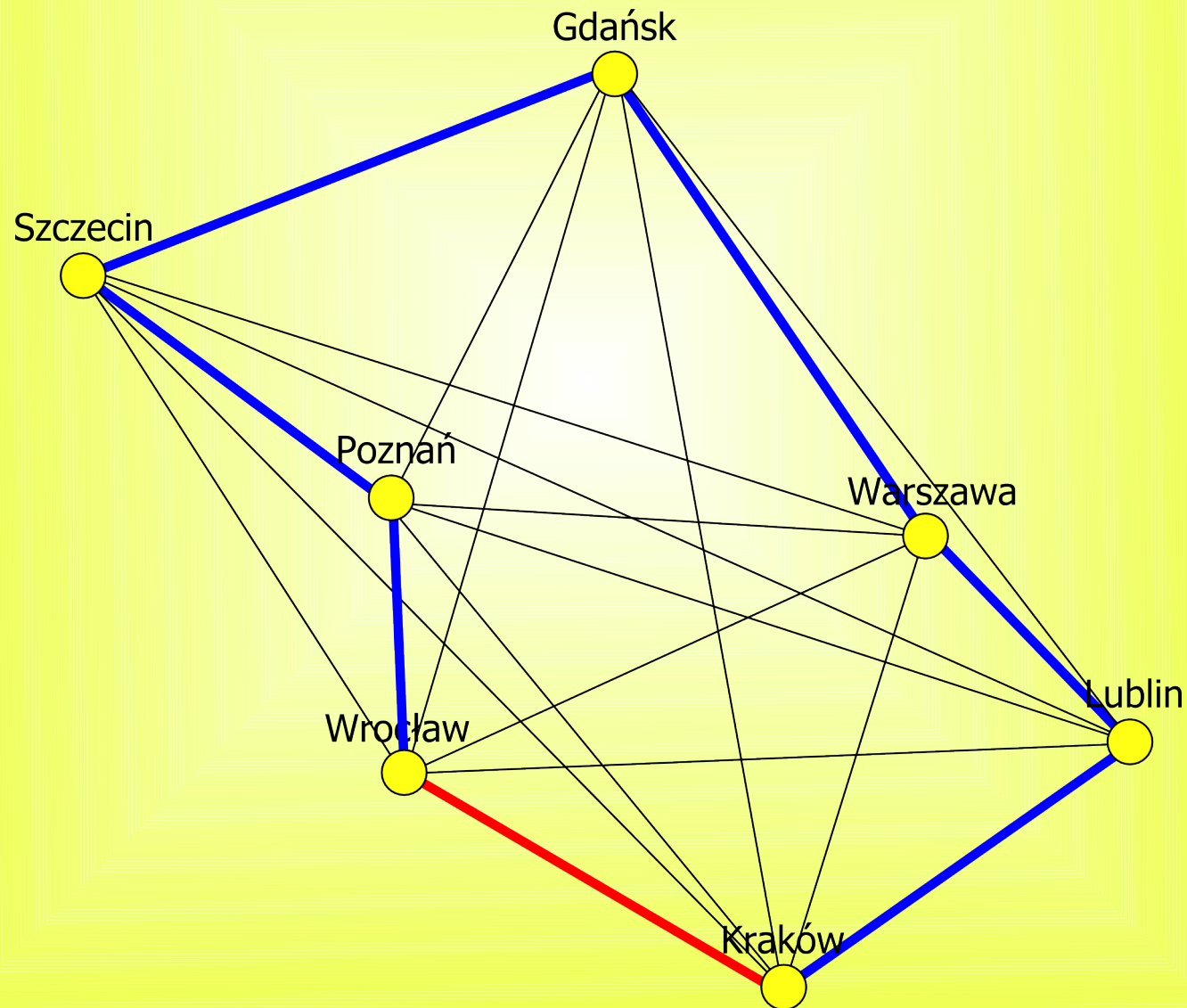
# Algorytm najkrótszej (lokalnie) ścieżki



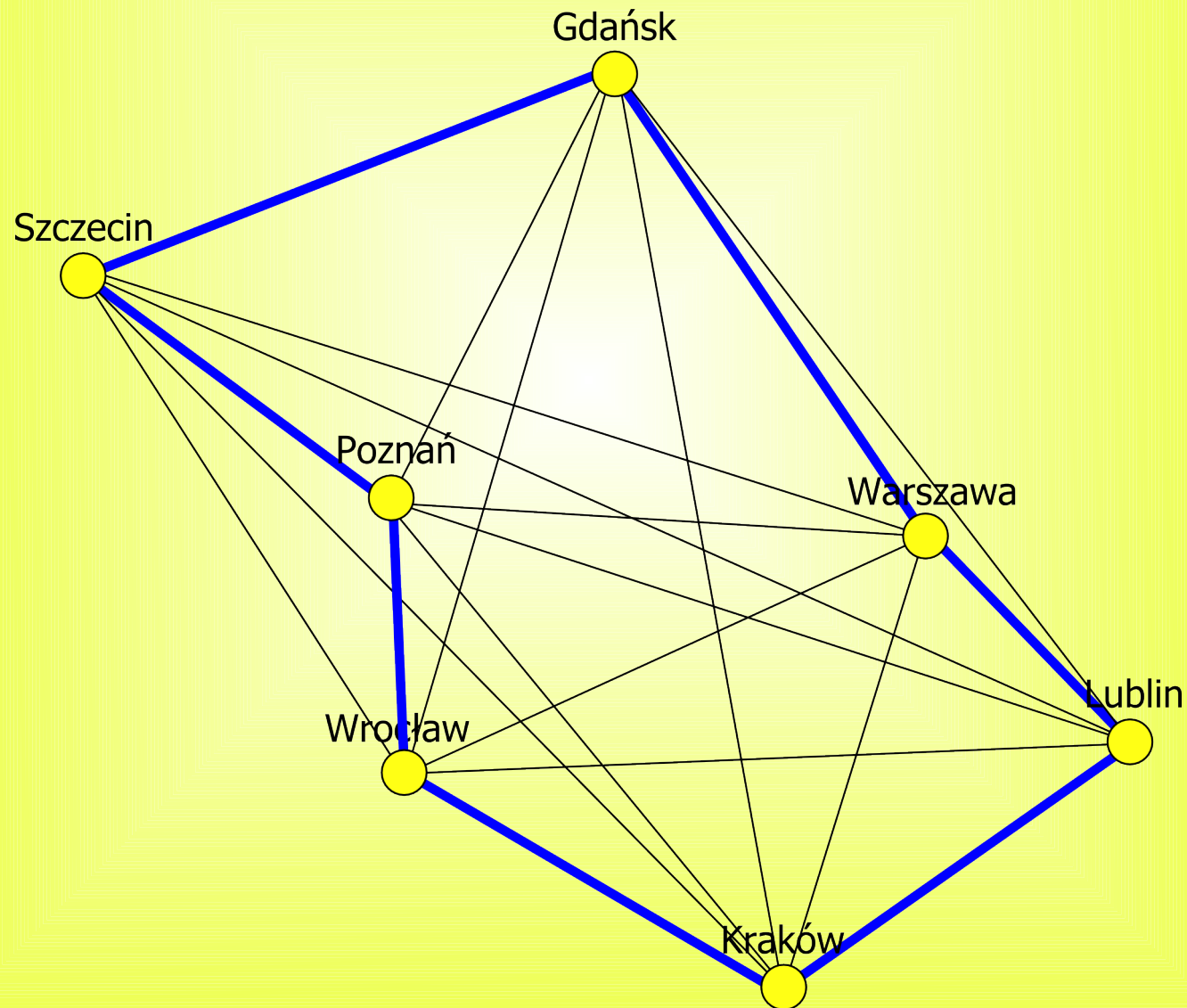
# Algorytm najkrótszej (lokalnie) ścieżki



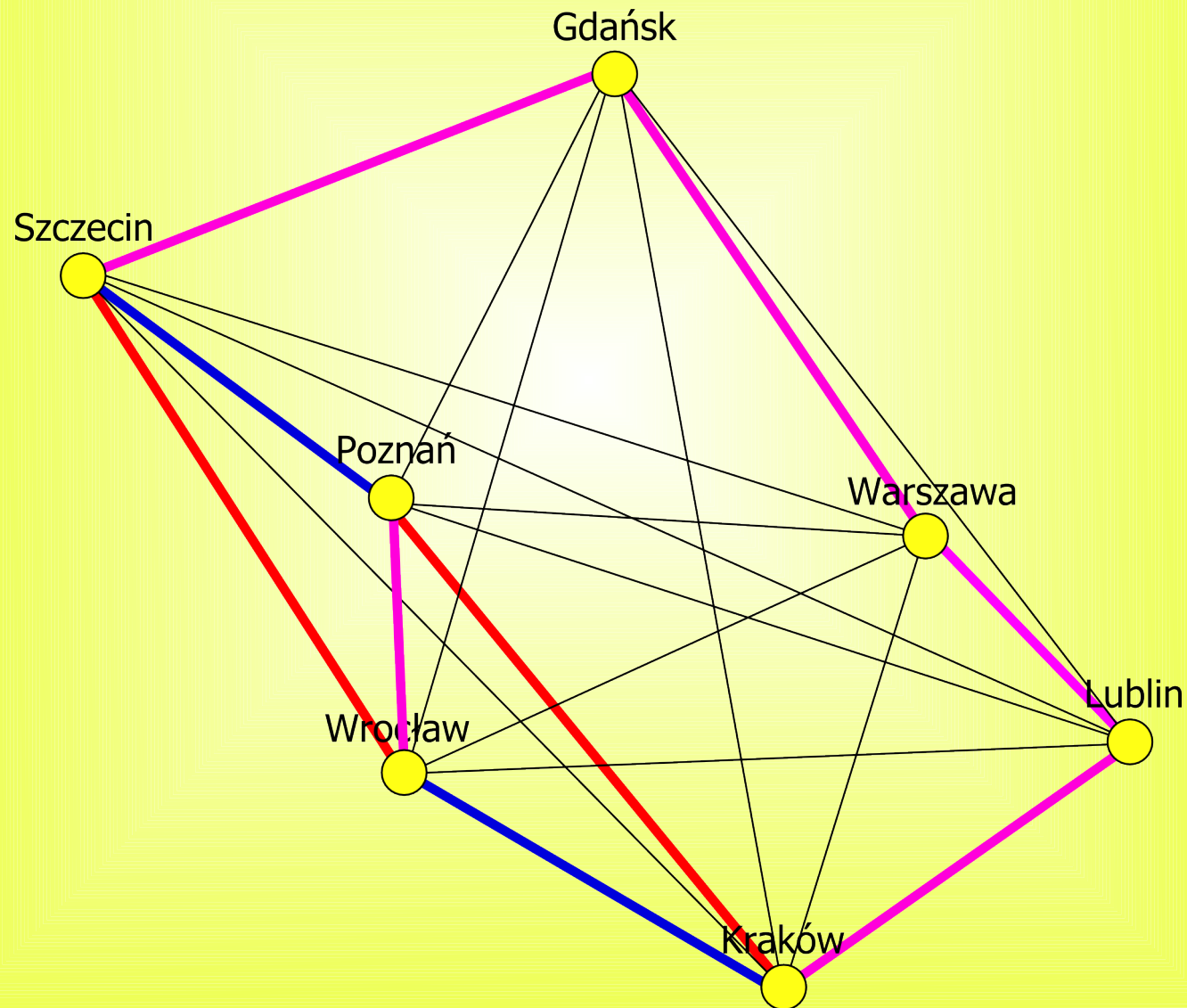
# Algorytm najkrótszej (lokalnie) ścieżki



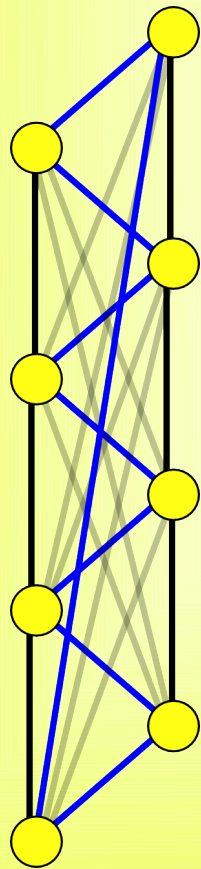
# Algorytm najkrótszej (lokalnie) ścieżki



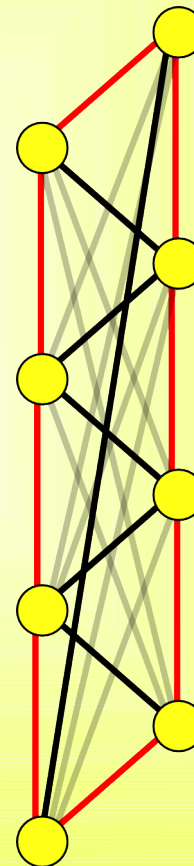
# Algorytm najkrótszej (lokalnie) ścieżki



# Algorytm najkrótszej (lokalnie) ścieżki nie zadziała np. dla tego przykładu



Suma dróg = 61



Suma dróg = 58



# Ocena

- Na 3,0
  - algorytm najkrótszej lokalnie ścieżki,
  - ORAZ NEH,
- Na 4,0
  - to, co na 3,0 + jedna metaheurystyka,
- Na 5,0
  - to, co na 3,0 + dwie metaheurystyki,
- +0,5 do +1,0 za GUI,
- -0,5 za każdy tydzień spóźnienia,
- +0,5 za wyeksportowanie algorytmu(-ów) do biblioteki DLL.