



**OBÓZ NAUKOWY
POLITECHNIKI WARSZAWSKIEJ**

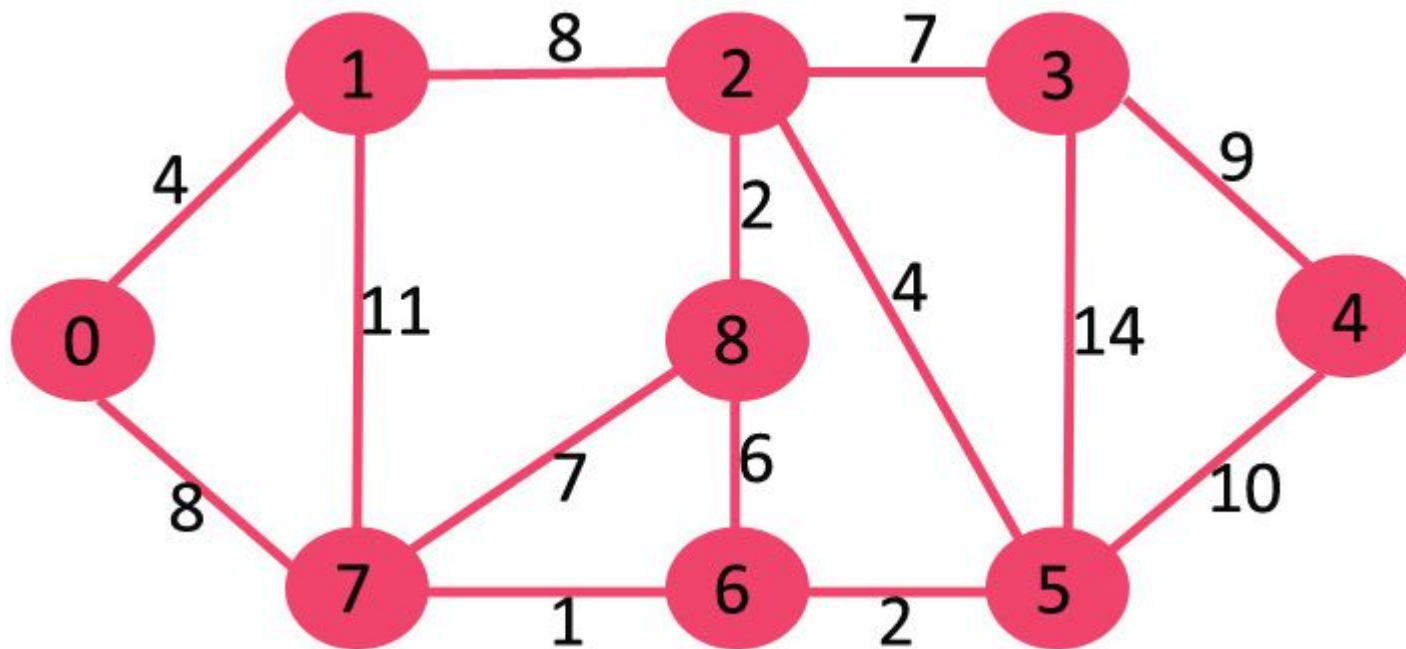
Pathfinding part 2

Dzień 3



Problem

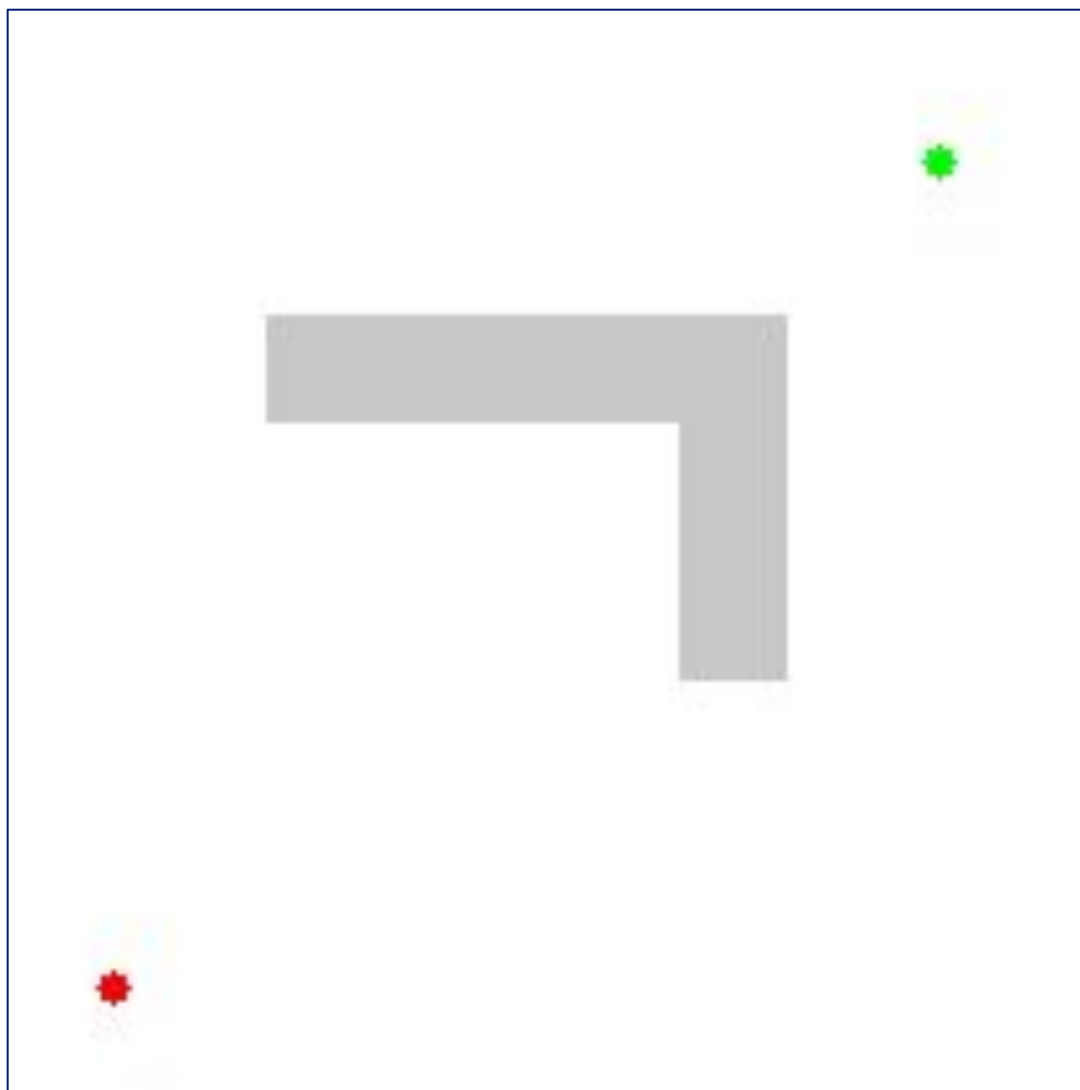
- chcemy się dostać od A do B jak najszybciej



Dijkstra

- znajduje odległości od wybranego wierzchołka do wszystkich innych
- działa tylko dla nieujemnych wag krawędzi
- opracowany na mojej uczelni :)





Intuicja

- Przeprowadzamy BFS i zapamiętujemy z którego wierzchołka przyszliśmy, oraz ile nas to kosztowało.



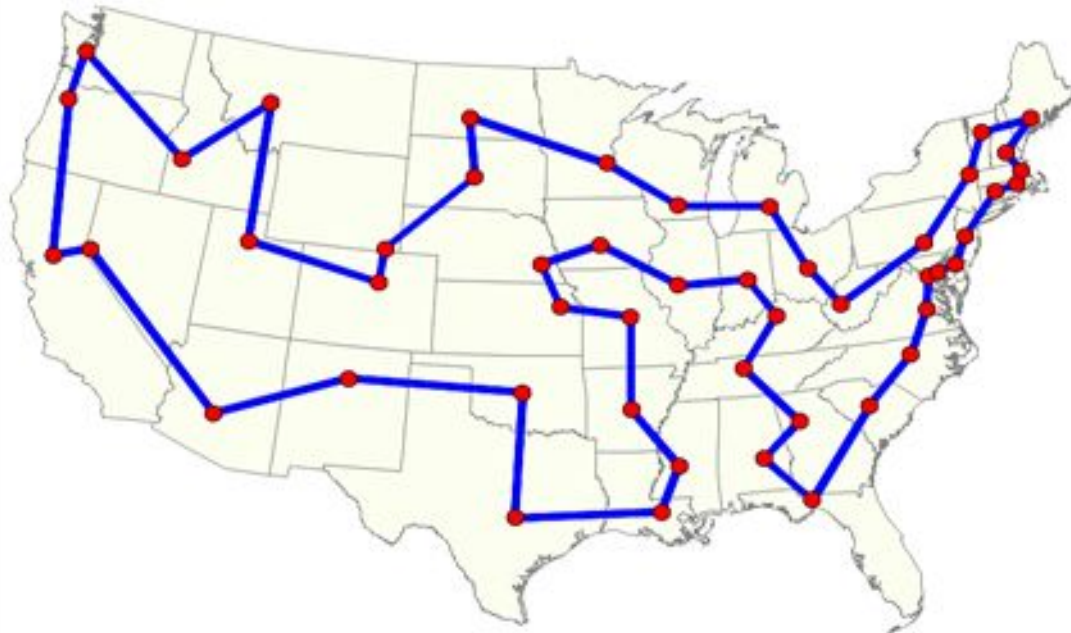
Pseudocode

1. Create a list of “distances” equal to the number of nodes and initialize each value to infinity
2. Set the “distance” to the starting node equal to 0
3. Create a list of “visited” nodes set to false for each node (since we haven’t visited any yet)
4. Loop through all the nodes
 - a. Loop through all the nodes again, and pick the one that is the shortest distance away *and* not yet visited
 - b. Set that node to visited
 - c. Set the distance in the distance list to the distance to that node
5. The original “distance” list should now contain the shortest distance to each node or infinity if a node is unreachable from the desired starting node



Problem komiwojaza

- Mając listę miast i odległości między nimi wszystkim podaj najkrótszą drogę przechodzącą przez wszystkie miasta



Inne zastosowania

- projektowanie układów scalonych
- sprawy logistyczne
- sekwencjonowanie DNA (miasta - kawałki DNA, odległości - podobieństwo między kawałkami)
- astronomia (jak najmniejsze przesunięcie teleskopu, żeby obejrzeć po kolei pewne gwiazdy)
-

