Politechnika Wrocławska

Autor: Filip Strózik 260377 Prowadzący: Mgr Inż. Michał Karol

Sztuczna Inteligencja i Inżynieria Wiedzy

Sprawozdanie z zadania 1

Wstęp teoretyczny i założenia

Celem algorytmów przeszukiwania Dijkstra i A* w tabelach i rozkładach komunikacji miejskiej jest znalezienie najkrótszej ścieżki lub najszybszego połączenia między dwoma punktami na sieci połączeń transportowych.

To co różni algorytm Dijkstry a algorytm A* to fakt, że A* wykorzystuje heurystykę, aby przyspieszyć proces przeszukiwania i znaleźć najkrótszą ścieżkę w bardziej złożonych grafach z różnymi kosztami przejścia. Wykorzystanie heurystyki, jest tylko możliwe przy posiadaniu dodatkowej wiedzy na temat wierzchołków.

W celu zastosowania tych algorytmów w tabelach i rozkładach komunikacji miejskiej, konieczne jest zbudowanie grafu, który będzie reprezentował sieć połączeń. Graf ten może być zbudowany z punktów przystanków jako wierzchołków, a krawędzie reprezentują połączenia między przystankami, przy czym koszt przejścia może być uzależniony od czasu przejazdu, odległości lub innych czynników

Budowanie grafu skierowanego i ważonego. Aby utrzymać realizm szukania najkrótszej ścieżki między przystankami utrzymujemy kierunkowość grafu. Wagą krawędzi jest czas przejazdu oraz ewentualnie czekanie na odjeżdżające połączenie.

Dla szybszego wykonywania i porównywania algorytmów postawiono na optymalizację budowanego grafu.

Łączymy przystanki o tej samej nazwie, mimo innych współrzędnych geograficznych, co daje nam małą ilość przystanków z ok 2000 na 935. Warto zwrócić uwagę, że końcowo po wczytaniu danych z pliku, współrzędne przystanku są pierwszymi współrzędnymi wczytanego przystanku z pliku co mogłoby mieć swoje konsekwencje. Ewentualnie można byłoby wyznaczyć współrzędne przystanku po środku wszystkich o tej samej nazwie.

Jeżeli łączymy przystanki to węzeł (przystanek) posiada wszystkie krawędzie wychodzące z niego w ciągu całego dnia, sprawia to, że mamy dostęp do większej liczby połączeń, aczkolwiek zwiększa to liczbę iteracji w przypadku przejścia po wszystkich krawędziach.

Dla typowego użytkownika, algorytmy będą zawsze oferować najkrótszy czas przejazdu, dlatego będąc na przystanku wchodzimy do pierwszego połączenia o najkrótszym czasie przejazdu do następnego przystanku, jeżeli jest to opłacalne czasowo.

Kolejną możliwością optymalizacyjną jest pobieranie połączeń wychodzących z przystanku w bramce czasowej wynoszącej określoną ilość minut. Jest to dość rozmyte założenie, ponieważ niektóre przystanki są częściej odwiedzane w ciągu dnia, niż inne, aczkolwiek bardzo przyśpiesza to szukanie. Oczywiście są przystanki, dla których w ciągu 30 minut nie przyjedzie nic, ale w naszym rozwiązaniu nie bierzemy tego pod uwagę, skoro skupiamy się na optymalizacji. Wtedy schodzimy z czasami przeszukiwań dla bardzo odległych przystanków do czasów < 100ms co już jest dość akceptowalnym czasem.

Technologia

Implementacja rozwiązana została napisana w C# 10.0 uruchamiana w .NET 6.

Implementacja

W ramach optymalizacji budujemy „Poprawnie” graf z tym, że upraszczamy przystanki o tej samej nazwie do jednego przystanku.

Naszym grafem nie jest wielka lista wszystkich możliwych krawędzi, tylko struktura węzłów (przystanków), gdzie każdy węzeł składa się z poniższych pól:

```
public string Name { get; set; }
public double Latitude { get; set; }
public double Longitude { get; set; }
public List<Edge> Edges { get; set; }
```

Każdy węzeł ma listę połączeń z niego wychodzących. O wiele bardziej to przyśpiesza liczenie kosztów użytych algorytmów, ponieważ nie musimy iterować po wszystkich krawędziach i liczyć na spełnienie warunków potencjalnego połączenia.

Krawędziami są połączenia, struktura jest zbliżona do pojedynczego wiersza z pliku connection_graph.csv, z tą różnicą, że mamy rzeczywistą referencje na węzły a nie tylko nazwę, co też wpływa na wydajność. (filtracja wszystkich krawędzi danego węzła)

```
public int Id { get; set; }
public string Company { get; set; }
public string Line { get; set; }
public TimeSpan ArrivalTime { get; set; }
public TimeSpan DepartureTime { get; set; }
public Node StartNode { get; set; }
public Node EndNode { get; set; }
```

Warto używać klasy TimeSpan, ponieważ pozwala ona na łatwe i precyzyjne wykonywanie operacji na czasie, a także oferuje wiele metod i właściwości do manipulacji czasem.

Dijkstra

```
var new_cost = cost_so_far[current] + graph.CalculateCost(current, next, currentTime);

if (!cost_so_far.ContainsKey(next.EndNode) || new_cost < cost_so_far[next.EndNode])
{
        cost_so_far[next.EndNode] = new_cost;
        var priority = new_cost;
        frontier.Enqueue(next.EndNode, priority);
        came_from[next.EndNode] = next;
}
```

A* time

```
var new_cost = cost_so_far[current] + graph.CalculateCost(current, next, currentTime);

if (!cost_so_far.ContainsKey(next.EndNode) || new_cost < cost_so_far[next.EndNode])
{
        cost_so_far[next.EndNode] = new_cost;
        var priority = new_cost + Graph.EuklidesHeuristic(endNode, next.EndNode);
        frontier.Enqueue(next.EndNode, priority);
        came_from[next.EndNode] = next;
}
```

A* changes

```
var new_cost = cost_so_far[current] + graph.CalculateCost(current, next, currentTime) +
Graph.LineChangeCost(came_from[current], next);

if (!cost_so_far.ContainsKey(next.EndNode) || new_cost < cost_so_far[next.EndNode])
{
        cost_so_far[next.EndNode] = new_cost;
        var priority = new_cost + Graph.EuklidesHeuristic(endNode, next.EndNode);
        frontier.Enqueue(next.EndNode, priority);
        came_from[next.EndNode] = next;
}
```

A* changes + strike

```
var new_cost = cost_so_far[current] + graph.CalculateCost(current, next, currentTime) +
Graph.LineChangeCost(came_from[current], next);
if (!cost_so_far.ContainsKey(next.EndNode) || new_cost < cost_so_far[next.EndNode])
{
        cost_so_far[next.EndNode] = new_cost;
        var priority = new_cost + Graph.EuklidesHeuristic(endNode, next.EndNode) + strikeCost;
        frontier.Enqueue(next.EndNode, priority);
        came_from[next.EndNode] = next;
}
```

Modyfikacja algorytmu A* ze zmianą linii, polega na dodaniu dodatkowej „kary” w przypadku potencjalnych zmian linii tuż po rozpoczęciu jazdy nową – „chcemy dłużej jechać linią, skoro już zdecydowaliśmy się na jakąkolwiek przesiadkę”. Kara maleje w raz z kolejnymi odwiedzonymi przystankami tej samej linii. Przyjąłem, że po odwiedzonych 12 przystanków tej samej linii, kary już nie ma. Lepsze wyniki daje tylko w przypadku dłuższych tras. Kod źródłowy zamieszczony w plikach. Dowody:

Dłuższe trasy		
Perzowa Jaworowa 12:30	Total time A* changes 72 Total visted nodes: 5700 Execution Time: 20 ms	Total time A* changes + strike 72 Total visted nodes: 4839 Execution Time: 23 ms
Jaworowa Perzowa 12:30	Total time A* changes 94 Total visted nodes: 21397 Execution Time: 73 ms	Total time A* changes + strike 64 Total visted nodes: 15131 Execution Time: 46 ms
Krótkie trasy		
DWORZEC NADODRZE DWORZEC GLOWNY 19:00	Total time A* changes 27 Total visted nodes: 1163 Execution Time: 5 ms	Total time A* changes + strike 27 Total visted nodes: 1310 Execution Time: 3 ms
Wyszynskiego Wielka 08:18	Total time A* changes 22 Total visted nodes: 1758 Execution Time: 10 ms	Total time A* changes + strike 22 Total visted nodes: 2867 Execution Time: 11 ms

Dla, krótszych tras gdzie zatraca się ważność „13” przystanków, działa mniej korzystnie, ponieważ dodaje dodatkowy koszt tam, gdzie rzeczywiście powinniśmy się przesiąść, połączenie ląduję dalej w kolejce priorytetowej i dłużej nam schodzi by trafić na dobrą zmianę linii.

Wyniki

Zadanie 1

Przykładowe wyniki algorytmów:

Perzowa Jaworowa 12:30	Magellana Rogowska 12:34	Wyszyńskiego Wielka 08:20	PL. GRUNWALDZKI DWORZEC GŁÓWNY 13:21
Total time Dijkstra 62 Total visted nodes: 11793 Execution Time: 59 ms	Total time Dijkstra 53 Total visted nodes: 9572 Execution Time: 78 ms	Total time Dijkstra 24 Total visted nodes: 7234 Execution Time: 56 ms	Total time Dijkstra 13 Total visted nodes: 1864 Execution Time: 48 ms
Total time A* time 62 Total visted nodes: 746 Execution Time: 12 ms	Total time A* time 53 Total visted nodes: 700 Execution Time: 16 ms	Total time A* time 24 Total visted nodes: 506 Execution Time: 10 ms	Total time A* time 13 Total visted nodes: 179 Execution Time: 13 ms
Total time A* changes 72 Total visted nodes: 5700 Execution Time: 24 ms	Total time A* changes 90 Total visted nodes: 11654 Execution Time: 37 ms	Total time A* changes 25 Total visted nodes: 863 Execution Time: 7 ms	Total time A* changes 13 Total visted nodes: 179 Execution Time: 8 ms
Total time A* changes + strike 72 Total visted nodes: 4839 Execution Time: 15 ms	Total time A* changes + strike 90 Total visted nodes: 9458 Execution Time: 26 ms	Total time A* changes + strike 25 Total visted nodes: 863 Execution Time: 9 ms	Total time A* changes + strike 13 Total visted nodes: 179 Execution Time: 8 ms

Przyjęte założenia też sprawiają, że atrakcyjniejszy czas jest lepszy niż koszt jednej przesiadki, przypominam, że wsiadamy do pierwszego, który daje najlepszy czas we wszystkich przypadkach algorytmów.

```
Run(g, "Wyszyńskiego", "Wielka", TimeSpan.Parse("08:18:00"));
```

```
------------------------
N       08:18:00        Wyszynskiego                    08:22:00        Ogród Botaniczny            4
N       08:22:00        Ogród Botaniczny                08:23:00        Katedra                     5
N       08:23:00        Katedra                         08:24:00        Urzad Wojewódzki (Muzeum N  6
N       08:24:00        Urzad Wojewódzki (Muzeum N      08:26:00        Poczta Główna               8
N       08:26:00        Poczta Główna                   08:29:00        GALERIA DOMINIKANSKA        11
N       08:29:00        GALERIA DOMINIKANSKA            08:33:00        DWORZEC GLOWNY              15
2       08:34:00        DWORZEC GLOWNY                  08:37:00        Arkady (Capitol)            19
2       08:37:00        Arkady (Capitol)                08:39:00        Zaolzianska                 21
2       08:39:00        Zaolzianska                     08:40:00        Wielka                      22
Total time Dijkstra 22
Total visted nodes: 5460
Execution Time: 69 ms
------------------------
N       08:18:00        Wyszynskiego                    08:22:00        Ogród Botaniczny            4
17      08:24:00        Ogród Botaniczny                08:26:00        pl. Bema                    8
17      08:26:00        pl. Bema                        08:28:00        Hala Targowa                10
17      08:28:00        Hala Targowa                    08:29:00        pl. Nowy Targ               11
17      08:29:00        pl. Nowy Targ                   08:31:00        GALERIA DOMINIKANSKA        13
17      08:31:00        GALERIA DOMINIKANSKA            08:33:00        Park Staromiejski           15
17      08:33:00        Park Staromiejski               08:35:00        Opera                       17
17      08:35:00        Opera                           08:39:00        Arkady (Capitol)            21
17      08:39:00        Arkady (Capitol)                08:41:00        Zaolzianska                 23
17      08:41:00        Zaolzianska                     08:42:00        Wielka                      24
Total time A* time 24
Total visted nodes: 549
Execution Time: 11 ms
------------------------
N       08:18:00        Wyszynskiego                    08:22:00        Ogród Botaniczny            4
N       08:22:00        Ogród Botaniczny                08:23:00        Katedra                     5
N       08:23:00        Katedra                         08:24:00        Urzad Wojewódzki (Muzeum N  6
N       08:24:00        Urzad Wojewódzki (Muzeum N      08:26:00        Poczta Główna               8
N       08:26:00        Poczta Główna                   08:29:00        GALERIA DOMINIKANSKA        11
N       08:29:00        GALERIA DOMINIKANSKA            08:33:00        DWORZEC GLOWNY              15
2       08:34:00        DWORZEC GLOWNY                  08:37:00        Arkady (Capitol)            119
2       08:37:00        Arkady (Capitol)                08:39:00        Zaolzianska                 121
2       08:39:00        Zaolzianska                     08:40:00        Wielka                      122
Total time A* changes 22
Total visted nodes: 1758
Execution Time: 13 ms
------------------------
N       08:18:00        Wyszynskiego                    08:22:00        Ogród Botaniczny            4
N       08:22:00        Ogród Botaniczny                08:23:00        Katedra                     5
N       08:23:00        Katedra                         08:24:00        Urzad Wojewódzki (Muzeum N  6
N       08:24:00        Urzad Wojewódzki (Muzeum N      08:26:00        Poczta Główna               8
N       08:26:00        Poczta Główna                   08:29:00        GALERIA DOMINIKANSKA        11
N       08:29:00        GALERIA DOMINIKANSKA            08:33:00        DWORZEC GLOWNY              15
2       08:34:00        DWORZEC GLOWNY                  08:37:00        Arkady (Capitol)            119
2       08:37:00        Arkady (Capitol)                08:39:00        Zaolzianska                 121
2       08:39:00        Zaolzianska                     08:40:00        Wielka                      122
Total time A* changes + strike 22
Total visted nodes: 2867
Execution Time: 19 ms
```

```
Run(g, "Wyszyńskiego", "Wielka", TimeSpan.Parse("08:19:00"));
```

```
-----------------------
70      08:21:00        Wyszynskiego                    08:22:00        Prusa                   3
70      08:22:00        Prusa                           08:24:00        Piastowska              5
70      08:24:00        Piastowska                      08:27:00        PL. GRUNWALDZKI         8
70      08:27:00        PL. GRUNWALDZKI                 08:28:00        most Grunwaldzki        9
149     08:28:00        most Grunwaldzki                08:30:00        Poczta Główna           11
149     08:30:00        Poczta Główna                   08:33:00        skwer Krasinskiego      14
149     08:33:00        skwer Krasinskiego              08:34:00        Wzgórze Partyzantów     15
149     08:34:00        Wzgórze Partyzantów             08:36:00        Renoma                  17
A       08:38:00        Renoma                          08:40:00        Arkady (Capitol)        21
6       08:41:00        Arkady (Capitol)                08:43:00        Zaolzianska             24
6       08:43:00        Zaolzianska                     08:44:00        Wielka                  25
Total time Dijkstra 25
Total visted nodes: 7234
Execution Time: 48 ms
-----------------------
70      08:22:00        Wyszynskiego                    08:24:00        Nowowiejska             5
11      08:24:00        Nowowiejska                     08:25:00        Jednosci Narodowej      6
11      08:25:00        Jednosci Narodowej              08:26:00        Na Szancach             7
11      08:26:00        Na Szancach                     08:28:00        pl. Bema                9
11      08:28:00        pl. Bema                        08:30:00        Hala Targowa            11
11      08:30:00        Hala Targowa                    08:31:00        pl. Nowy Targ           12
11      08:31:00        pl. Nowy Targ                   08:33:00        GALERIA DOMINIKANSKA    14
11      08:33:00        GALERIA DOMINIKANSKA            08:35:00        Wzgórze Partyzantów     16
11      08:35:00        Wzgórze Partyzantów             08:37:00        DWORZEC GLOWNY          18
11      08:37:00        DWORZEC GLOWNY                  08:40:00        Arkady (Capitol)        21
6       08:41:00        Arkady (Capitol)                08:43:00        Zaolzianska             24
6       08:43:00        Zaolzianska                     08:44:00        Wielka                  25
Total time A* time 25
Total visted nodes: 506
Execution Time: 10 ms
-----------------------
70      08:21:00        Wyszynskiego                    08:22:00        Prusa                   3
70      08:22:00        Prusa                           08:24:00        Piastowska              5
70      08:24:00        Piastowska                      08:27:00        PL. GRUNWALDZKI         8
70      08:27:00        PL. GRUNWALDZKI                 08:28:00        most Grunwaldzki        9
70      08:28:00        most Grunwaldzki                08:30:00        Urzad Wojewódzki (Impart)  11
70      08:30:00        Urzad Wojewódzki (Impart)       08:33:00        pl. Wróblewskiego       14
70      08:33:00        pl. Wróblewskiego               08:34:00        Komuny Paryskiej        15
70      08:34:00        Komuny Paryskiej                08:35:00        Kosciuszki              16
70      08:35:00        Kosciuszki                      08:37:00        Pulaskiego              18
70      08:37:00        Pulaskiego                      08:39:00        DWORZEC GLOWNY          20
70      08:39:00        DWORZEC GLOWNY                  08:42:00        Arkady (Capitol)        23
70      08:42:00        Arkady (Capitol)                08:44:00        Zaolzianska             25
70      08:44:00        Zaolzianska                     08:45:00        Wielka                  26
Total time A* changes 26
Total visted nodes: 863
Execution Time: 10 ms
-----------------------
70      08:21:00        Wyszynskiego                    08:22:00        Prusa                   3
70      08:22:00        Prusa                           08:24:00        Piastowska              5
70      08:24:00        Piastowska                      08:27:00        PL. GRUNWALDZKI         8
70      08:27:00        PL. GRUNWALDZKI                 08:28:00        most Grunwaldzki        9
70      08:28:00        most Grunwaldzki                08:30:00        Urzad Wojewódzki (Impart)  11
70      08:30:00        Urzad Wojewódzki (Impart)       08:33:00        pl. Wróblewskiego       14
70      08:33:00        pl. Wróblewskiego               08:34:00        Komuny Paryskiej        15
70      08:34:00        Komuny Paryskiej                08:35:00        Kosciuszki              16
70      08:35:00        Kosciuszki                      08:37:00        Pulaskiego              18
70      08:37:00        Pulaskiego                      08:39:00        DWORZEC GLOWNY          20
70      08:39:00        DWORZEC GLOWNY                  08:42:00        Arkady (Capitol)        23
70      08:42:00        Arkady (Capitol)                08:44:00        Zaolzianska             25
70      08:44:00        Zaolzianska                     08:45:00        Wielka                  26
Total time A* changes + strike 26
Total visted nodes: 863
Execution Time: 8 ms
```

Porównajmy wyniki działania algorytmów A* changes. Mimo braku przesiadek w podjęciu się linii 70 o godzinie 08:21, dojeżdżamy o 08:45 na miejsce, czas trwania podróży to 26 minut.

Mimo, tego, że mamy dostępny tramwaj 70 o 08:21 to wybieramy wcześniejszy, dzięki któremu jesteśmy szybciej w miejscu docelowym.

Zadanie 2

Dla listy przystanków np.

```
List<string> points = new List<string> { "PL. GRUNWALDZKI", "Hutmen",
"Dmowskiego", "Broniewskiego", "Jaworowa" };
```

Oraz limitu iteracji: 120 czas

```
----------------

Best time Tabu 124
------------------------
6       10:04:00        Wita Stwosza                    10:05:00        Uniwersytecka           5
6       10:06:00        Uniwersytecka                   10:09:00        Dubois                  9
15      10:09:00        Dubois                          10:11:00        Pomorska                11
15      10:11:00        Pomorska                        10:13:00        Dmowskiego              13
Total time A* time 13
Total visted nodes: 65
------------------------
C       10:13:00        Dmowskiego                      10:15:00        Pomorska                2
142     10:19:00        Pomorska                        10:21:00        pl. Strzelecki          8
142     10:21:00        pl. Strzelecki                  10:23:00        Kleczkowska             10
142     10:23:00        Kleczkowska                     10:25:00        most Osobowicki         12
129     10:25:00        most Osobowicki                 10:27:00        Baltycka                14
129     10:27:00        Baltycka                        10:30:00        Broniewskiego           17
Total time A* time 17
Total visted nodes: 101
------------------------
111     10:31:00        Broniewskiego                   10:33:00        Zakladowa               3
111     10:33:00        Zakladowa                       10:35:00        Slonimskiego            5
111     10:35:00        Slonimskiego                    10:38:00        Daszynskiego            8
23      10:38:00        Daszynskiego                    10:40:00        Nowowiejska             10
70      10:44:00        Nowowiejska                     10:46:00        Wyszynskiego            16
70      10:46:00        Wyszynskiego                    10:47:00        Prusa                   17
70      10:47:00        Prusa                           10:49:00        Piastowska              19
70      10:49:00        Piastowska                      10:52:00        PL. GRUNWALDZKI         22
Total time A* time 22
Total visted nodes: 161
------------------------
C       10:52:00        PL. GRUNWALDZKI                 10:55:00        Katedra                 3
C       10:55:00        Katedra                         10:57:00        pl. Bema                5
C       10:57:00        pl. Bema                        11:00:00        Dubois                  8
C       11:00:00        Dubois                          11:03:00        Dmowskiego              11
74      11:06:00        Dmowskiego                      11:08:00        PL. JANA PAWLA II       16
74      11:08:00        PL. JANA PAWLA II               11:11:00        pl. Orlat Lwowskich     19
74      11:11:00        pl. Orlat Lwowskich             11:13:00        pl. Legionów            21
74      11:13:00        pl. Legionów                    11:14:00        Kolejowa                22
74      11:14:00        Kolejowa                        11:15:00        Grabiszynska            23
74      11:15:00        Grabiszynska                    11:17:00        Pereca                  25
74      11:17:00        Pereca                          11:19:00        Stalowa                 27
74      11:19:00        Stalowa                         11:20:00        pl. Srebrny             28
74      11:20:00        pl. Srebrny                     11:21:00        Bzowa (Centrum Zajezdnia)  29
74      11:21:00        Bzowa (Centrum Zajezdnia)       11:22:00        Hutmen                  30
Total time A* time 30
Total visted nodes: 1370
------------------------
124     11:23:00        Hutmen                          11:25:00        Bzowa (Centrum Zajezdnia)  3
124     11:25:00        Bzowa (Centrum Zajezdnia)       11:26:00        pl. Srebrny             4
74      11:27:00        pl. Srebrny                     11:28:00        Stalowa                 6
124     11:28:00        Stalowa                         11:29:00        Grochowa                7
124     11:29:00        Grochowa                        11:30:00        Krucza                  8
124     11:30:00        Krucza                          11:33:00        Rondo                   11
20      11:33:00        Rondo                           11:34:00        Sztabowa                12
20      11:34:00        Sztabowa                        11:36:00        Hallera                 14
127     11:36:00        Hallera                         11:37:00        Sudecka                 15
127     11:37:00        Sudecka                         11:39:00        Wisniowa                17
127     11:39:00        Wisniowa                        11:40:00        Jaworowa                18
Total time A* time 18
Total visted nodes: 342
------------------------
127     11:42:00        Jaworowa                        11:44:00        Wisniowa                4
15      11:45:00        Wisniowa                        11:47:00        Uniwersytet Ekonomiczny 7
15      11:47:00        Uniwersytet Ekonomiczny         11:48:00        Sanocka                 8
15      11:48:00        Sanocka                         11:51:00        DWORZEC AUTOBUSOWY      11
15      11:51:00        DWORZEC AUTOBUSOWY              11:53:00        Dworzec Główny (Stawowa)  13
15      11:53:00        Dworzec Główny (Stawowa)        11:55:00        Arkady (Capitol)        15
6       11:57:00        Arkady (Capitol)                11:59:00        Renoma                  19
6       11:59:00        Renoma                          12:00:00        Opera                   20
6       12:00:00        Opera                           12:01:00        SWIDNICKA (Dom Europy)  21
6       12:01:00        SWIDNICKA (Dom Europy)          12:03:00        Olawska                 23
6       12:03:00        Olawska                         12:04:00        Wita Stwosza            24
Total time A* time 24
Total visted nodes: 225

----------------
```

A dla zmian przystanków

```
Best time Tabu 130
------------------------
6       10:04:00        Wita Stwosza                    10:05:00        Uniwersytecka               5
6       10:06:00        Uniwersytecka                   10:09:00        Dubois                      9
15      10:09:00        Dubois                          10:11:00        Pomorska                    111
15      10:11:00        Pomorska                        10:13:00        Dmowskiego                  113
Total time A* changes 13
Total visted nodes: 1403
------------------------
15      10:13:00        Dmowskiego                      10:15:00        PL. JANA PAWLA II           2
15      10:15:00        PL. JANA PAWLA II               10:18:00        pl. Orlat Lwowskich         5
15      10:18:00        pl. Orlat Lwowskich             10:20:00        pl. Legionów                7
11      10:23:00        pl. Legionów                    10:24:00        Kolejowa                    111
11      10:24:00        Kolejowa                        10:25:00        Grabiszynska                112
11      10:25:00        Grabiszynska                    10:27:00        Pereca                      114
11      10:27:00        Pereca                          10:29:00        Stalowa                     116
11      10:29:00        Stalowa                         10:30:00        pl. Srebrny                 117
11      10:30:00        pl. Srebrny                     10:31:00        Bzowa (Centrum Zajezdnia)   118
11      10:31:00        Bzowa (Centrum Zajezdnia)       10:32:00        Hutmen                      119
Total time A* changes 19
Total visted nodes: 1288
------------------------
11      10:32:00        Hutmen                          10:34:00        FAT                         2
20      10:38:00        FAT                             10:40:00        Aleja Pracy                 108
20      10:40:00        Aleja Pracy                     10:41:00        Ojca Beyzyma                109
20      10:41:00        Ojca Beyzyma                    10:43:00        Mielecka                    111
20      10:43:00        Mielecka                        10:44:00        Gajowicka                   112
20      10:44:00        Gajowicka                       10:46:00        Hallera                     114
127     10:49:00        Hallera                         10:50:00        Sudecka                     218
127     10:50:00        Sudecka                         10:52:00        Wisniowa                    220
127     10:52:00        Wisniowa                        10:53:00        Jaworowa                    221
Total time A* changes 21
Total visted nodes: 6120
------------------------
9       10:53:00        Jaworowa                        10:54:00        Wisniowa                    1
9       10:54:00        Wisniowa                        10:56:00        Uniwersytet Ekonomiczny     3
9       10:56:00        Uniwersytet Ekonomiczny         10:57:00        Sanocka                     4
9       10:57:00        Sanocka                         11:00:00        DWORZEC AUTOBUSOWY          7
9       11:00:00        DWORZEC AUTOBUSOWY              11:02:00        DWORZEC GLOWNY              9
9       11:02:00        DWORZEC GLOWNY                  11:04:00        Wzgórze Partyzantów         11
9       11:04:00        Wzgórze Partyzantów             11:07:00        GALERIA DOMINIKANSKA        14
9       11:07:00        GALERIA DOMINIKANSKA            11:08:00        pl. Nowy Targ               15
9       11:08:00        pl. Nowy Targ                   11:09:00        Hala Targowa                16
9       11:09:00        Hala Targowa                    11:12:00        pl. Bema                    19
6       11:12:00        pl. Bema                        11:13:00        Na Szancach                 120
6       11:13:00        Na Szancach                     11:14:00        Jednosci Narodowej          121
6       11:14:00        Jednosci Narodowej              11:16:00        Nowowiejska                 123
1       11:18:00        Nowowiejska                     11:20:00        Slowianska                  227
1       11:20:00        Slowianska                      11:22:00        DWORZEC NADODRZE            229
1       11:23:00        DWORZEC NADODRZE                11:25:00        Trzebnicka                  232
1       11:25:00        Trzebnicka                      11:28:00        Broniewskiego               235
Total time A* changes 35
Total visted nodes: 6201
------------------------
1       11:28:00        Broniewskiego                   11:31:00        Trzebnicka                  3
1       11:31:00        Trzebnicka                      11:33:00        DWORZEC NADODRZE            5
1       11:34:00        DWORZEC NADODRZE                11:36:00        Slowianska                  8
1       11:36:00        Slowianska                      11:38:00        Nowowiejska                 10
1       11:38:00        Nowowiejska                     11:40:00        Wyszynskiego                12
1       11:40:00        Wyszynskiego                    11:41:00        Prusa                       13
1       11:41:00        Prusa                           11:43:00        Piastowska                  15
1       11:43:00        Piastowska                      11:46:00        PL. GRUNWALDZKI             18
Total time A* changes 18
Total visted nodes: 273
------------------------
33      11:46:00        PL. GRUNWALDZKI                 11:47:00        most Grunwaldzki            1
33      11:47:00        most Grunwaldzki                11:49:00        Urzad Wojewódzki (Impart)   3
33      11:49:00        Urzad Wojewódzki (Impart)       11:53:00        GALERIA DOMINIKANSKA        7
17      12:01:00        GALERIA DOMINIKANSKA            12:03:00        Park Staromiejski           117
17      12:03:00        Park Staromiejski               12:05:00        Opera                       119
7       12:06:00        Opera                           12:07:00        SWIDNICKA (Dom Europy)      221
7       12:07:00        SWIDNICKA (Dom Europy)          12:09:00        Olawska                     223
7       12:09:00        Olawska                         12:10:00        Wita Stwosza                224
Total time A* changes 24
Total visted nodes: 10713
```

Miniejszy przykład:

```
List<string> points = new List<string> { "PL. GRUNWALDZKI", "Hutmen", "Dmowskiego"};
```

Maksymalnie iteracji: 3

```
Best time Tabu 79
-----------------------
6       10:04:00        Wita Stwosza                    10:05:00        Uniwersytecka               5
6       10:06:00        Uniwersytecka                   10:09:00        Dubois                      9
6       10:09:00        Dubois                          10:12:00        pl. Bema                    12
9       10:12:00        pl. Bema                        10:14:00        Ogród Botaniczny            14
9       10:14:00        Ogród Botaniczny                10:15:00        Górnickiego                 15
9       10:15:00        Górnickiego                     10:16:00        Piastowska                  16
70      10:19:00        Piastowska                      10:22:00        PL. GRUNWALDZKI             22
Total time A* time 22
Total visted nodes: 126
-----------------------
C       10:22:00        PL. GRUNWALDZKI                 10:25:00        Katedra                     3
C       10:25:00        Katedra                         10:27:00        pl. Bema                    5
C       10:27:00        pl. Bema                        10:30:00        Dubois                      8
C       10:30:00        Dubois                          10:33:00        Dmowskiego                  11
Total time A* time 11
Total visted nodes: 162
-----------------------
74      10:36:00        Dmowskiego                      10:38:00        PL. JANA PAWLA II           5
74      10:38:00        PL. JANA PAWLA II               10:41:00        pl. Orlat Lwowskich         8
74      10:41:00        pl. Orlat Lwowskich             10:43:00        pl. Legionów                10
74      10:43:00        pl. Legionów                    10:44:00        Kolejowa                    11
74      10:44:00        Kolejowa                        10:45:00        Grabiszynska                12
74      10:45:00        Grabiszynska                    10:47:00        Pereca                      14
74      10:47:00        Pereca                          10:49:00        Stalowa                     16
74      10:49:00        Stalowa                         10:50:00        pl. Srebrny                 17
74      10:50:00        pl. Srebrny                     10:51:00        Bzowa (Centrum Zajezdnia)   18
74      10:51:00        Bzowa (Centrum Zajezdnia)       10:52:00        Hutmen                      19
Total time A* time 19
Total visted nodes: 1201
-----------------------
124     10:53:00        Hutmen                          10:55:00        Bzowa (Centrum Zajezdnia)   3
124     10:55:00        Bzowa (Centrum Zajezdnia)       10:56:00        pl. Srebrny                 4
74      10:57:00        pl. Srebrny                     10:58:00        Stalowa                     6
74      10:58:00        Stalowa                         11:00:00        Pereca                      8
74      11:00:00        Pereca                          11:01:00        Grabiszynska                9
74      11:01:00        Grabiszynska                    11:02:00        Kolejowa                    10
74      11:02:00        Kolejowa                        11:04:00        pl. Legionów                12
74      11:04:00        pl. Legionów                    11:06:00        pl. Orlat Lwowskich         14
148     11:06:00        pl. Orlat Lwowskich             11:08:00        Renoma                      16
17      11:09:00        Renoma                          11:10:00        Opera                       18
6       11:15:00        Opera                           11:16:00        SWIDNICKA (Dom Europy)      24
6       11:16:00        SWIDNICKA (Dom Europy)          11:18:00        Olawska                     26
6       11:18:00        Olawska                         11:19:00        Wita Stwosza                27
Total time A* time 27
Total visted nodes: 557
```

```
Best time Tabu 85
------------------------
6       10:04:00        Wita Stwosza                    10:05:00        Uniwersytecka           5
6       10:06:00        Uniwersytecka                   10:09:00        Dubois                  9
6       10:09:00        Dubois                          10:12:00        pl. Bema                12
C       10:19:00        pl. Bema                        10:22:00        Katedra                 122
C       10:22:00        Katedra                         10:25:00        PL. GRUNWALDZKI         125
Total time A* changes 25
Total visted nodes: 1461
------------------------
111     10:25:00        PL. GRUNWALDZKI                 10:27:00        Reja                    2
111     10:27:00        Reja                            10:29:00        Katedra                 4
111     10:29:00        Katedra                         10:31:00        Ogród Botaniczny        6
111     10:31:00        Ogród Botaniczny                10:33:00        Swietokrzyska           8
111     10:33:00        Swietokrzyska                   10:34:00        pl. Bema                9
6       10:35:00        pl. Bema                        10:37:00        Dubois                  112
144     10:37:00        Dubois                          10:39:00        Pomorska                214
144     10:39:00        Pomorska                        10:41:00        Dmowskiego              216
Total time A* changes 16
Total visted nodes: 6891
------------------------
144     10:41:00        Dmowskiego                      10:43:00        PL. JANA PAWLA II       2
144     10:43:00        PL. JANA PAWLA II               10:45:00        pl. Orlat Lwowskich     4
144     10:45:00        pl. Orlat Lwowskich             10:47:00        pl. Legionów            6
144     10:47:00        pl. Legionów                    10:49:00        Pilsudskiego            8
144     10:49:00        Pilsudskiego                    10:51:00        Zielinskiego            10
144     10:51:00        Zielinskiego                    10:53:00        Zaporoska               12
144     10:53:00        Zaporoska                       10:54:00        Krucza                  13
124     10:57:00        Krucza                          10:59:00        Grochowa                118
124     10:59:00        Grochowa                        11:01:00        Stalowa                 120
124     11:01:00        Stalowa                         11:02:00        pl. Srebrny             121
124     11:02:00        pl. Srebrny                     11:03:00        Bzowa (Centrum Zajezdnia)  122
124     11:03:00        Bzowa (Centrum Zajezdnia)       11:04:00        Hutmen                  123
Total time A* changes 23
Total visted nodes: 1678
------------------------
5       11:06:00        Hutmen                          11:07:00        Bzowa (Centrum Zajezdnia)  3
5       11:07:00        Bzowa (Centrum Zajezdnia)       11:08:00        pl. Srebrny             4
5       11:08:00        pl. Srebrny                     11:09:00        Stalowa                 5
5       11:09:00        Stalowa                         11:11:00        Pereca                  7
5       11:11:00        Pereca                          11:12:00        Grabiszynska            8
5       11:12:00        Grabiszynska                    11:13:00        Kolejowa                9
5       11:13:00        Kolejowa                        11:15:00        pl. Legionów            11
5       11:15:00        pl. Legionów                    11:17:00        Arkady (Capitol)        13
7       11:18:00        Arkady (Capitol)                11:20:00        Renoma                  116
7       11:20:00        Renoma                          11:21:00        Opera                   117
7       11:21:00        Opera                           11:22:00        SWIDNICKA (Dom Europy)  118
7       11:22:00        SWIDNICKA (Dom Europy)          11:24:00        Olawska                 120
7       11:24:00        Olawska                         11:25:00        Wita Stwosza            121
Total time A* changes 21
Total visted nodes: 1092
```

Sukces znalezienia najlepszego rozwiązania polega na dobraniu dobrej ilości limitu iteracji, co jest związane z istotą dobrego dobierania sąsiadów rozwiązania. Zapewniając najlepsze rozwiązanie byłby to wszystkie permutacje, lecz wtedy mamy sprawdzamy n! kombinacji, n elementowego zbioru przystanków, co mija się z celem.

Wnioski

Dość dużym problemem, początkowo przy zadaniu pierwszym było „ważenie” kosztów, tak aby dawały zadowalające efekty.

Mnogość założeń jakie można stworzyć i przypadków skrajnych przytłacza, aczkolwiek zwiększa kreatywność rozwiązań.

Warto wybrać silnie typizowany język, ponieważ, deklaracje typów pomagają rozjaśnić co się dzieje w kodzie.

Jestem dość zadowolony z optymalizacji oraz budowy samego grafu.

Podsumowując zadanie pierwsze. Dijkstra da najlepsze rozwiązanie dużym kosztem, A * da najszybciej dobre rozwiązanie, zależnie od heurystyki. W przypadku kryterium przesiadek, wynik jest zadowalający przy ustalonych założeniach. Modyfikacja byłaby pewnie używana, gdy wstępnie stwierdzilibyśmy, że odległość między punktami jest wystarczająco duża, by użyć zmodyfikowanej wersji.

W zadaniu drugim, ciężko początkowo zrozumieć jak generować sąsiedztwo aktualnego rozwiązania, u mnie jest to generowanie listy rozwiązań mającej rozmiar rozwiązania, czyli dla n elementów mam n nowych rozwiązań, co w przypadku blokowania rozwiązań za pomocą tabu, można dojść do momentu, gdzie zbiór sąsiedztw będzie pusty, dlatego takich momentach, kompletnie zwalniam listę tabu, aby możliwie znaleźć najlepsze rozwiązanie. To założenie, sprawia, że zawsze wykonujemy maksymalną liczbę iteracji, jednak zwiększa szanse na znalezienie optimum globalnego.