Wyznaczenie sumarycznie najkrótszych ścieżek rozłącznych krawędziowo
Sprawozdanie 3

Korzeniowski Wojciech, Gadawski Łukasz

2 czerwca 2015

1 Cel projektu

Realizacja projektu polegała na zaimplementowaniu zmodyfikowanej wersji algorytmu Dijkstry, która umożliwia znalezienie dwóch rozłącznych krawędziowo ścieżek w skierowanym grafie dla dowolnej pary wierzchołków.

Jednym z praktycznych zastosowań wyznaczanie najkrótszych rozłącznych krawędziowo ścieżek w grafie może być chęć poprawy niezawodności sieci przesyłowych poprzez wykorzystanie najbardziej optymalnych ścieżek jako zapasowyh ścieżek. Innym z zastosowaniem może być podział transmisji danych pomiędzy dwie ścieżki dzięki podczas wystąpienia awari możliwa jest częściowa transmisja danych.

2 Struktury danych

W programie wykorzystane są następujące struktury:

- Vertex - reprezentuje wierzchołek, zawiera nazwę.
- Edge - reprezentuje krawędź skierowaną, zawiera wierzchołek początkowy, wierzchołek końcowy, wagę krawędzi.
- Graph - reprezentuje graf, zawiera zbiór krawędzi oraz zbiór wierzchołków należących do grafu.
- GraphPath - reprezentuje ścieżkę w grafie, zawiera listę krawędzi.

3 Opis algorytmu

Poniżej przedstawiony został przebieg wykonania algorytmu w celu znalezienie dwóch rozłącznych krawędziowo ścieżek.

1. Znajdź najkrótszą ścieżkę przy pomocy standardowego algorytmu Dijkstry.
2. Dla każdej krawędzi znalezionej ścieżki odwróć jej zwrot oraz zmień wagę na ujemną.

3. Ponownie znajdź najkrótszą ścieżkę w tak otrzymanym grafie, tym razem korzystając ze zmodyfikowanej wersji algorytmu Dijkstry. (opis 3.1)

4. Ze zbioru reprezentującego sumę zbiorów krawędzi znalezionej ścieżki usuń część wspólną.

5. Z pozostałego zbioru krawędzi znajdź iteracyjnie ścieżki idąc od wierzchołka końcowego ku początkowemu usuwając przy tym krawędzie zużyte z wynikowego zbioru.

6. Zwróc uzyskane ścieżki.

3.1 Zmodyfikowany algorytm Dijkstry

Oznaczenia:

S - wierzchołek startowy
D - wierzchołek docelowy
$neighbour(S)$ - zbiór wierzchołków sąsiadujących z S
$distance(A)$ - odległość od wierzchołka startowego S do wierzchołka A
$predecessor(A)$ - tablica zawierająca poprzednik wierzchołka A dla znalezionej ścieżki
$cisited$ - zbiór wierzchołków odwiedzonych przez algorytm
$searchNodes$ - zbiór wierzchołków do wyszukania najkrótszej ścieżki

Kroki algorytm:

1. Dla każdego wierzchołka $V \in v$ przypisz $distance(V) = +\inf$.

2. Dla wierzchołka startowego przypisz $distance(S) = 0$.

3. Dla każdego $NS \in neighbours(S)$ przypisz $distance(NS) = weight(S, NS)$ oraz $predecessor(NS) = S$.

4. Dodaj sąsiadów wierzchołka S do zbioru wierzchołków do przeszukania $searchNodes.add(neighbour(S))$.

5. Znajdź $F \in searchNodess$ dla którego wartość $distance(F)$ jest najminejsza.

6. Usuń wierzchołek F ze zbioru wierzchołków do przeszukania $searchNodes.remove(F)$.

7. Jeżeli $F = D$ to ZAKOŃCZ

8. Dla każdego wierzchołka $NF \in neighbour(F)$ jeżeli $distance(F)+weight(F, NF) < distance(NF)$ przypisz nową, krótszą odległość $distance(ND) = distance(F)+ weight(F, NF)$ oraz nowego poprzednika $predecessor(NF) = F$ oraz dodaj wierzchołek do zbioru przeszukania $searchNodes.add(NF)$.

4 Przebieg działania aplikacji

W ramach projektu została stworzona aplikacja udostępniające następujące funkcjonalności.

4.1 Wykonanie na podstawie danych z pliku tekstowego

Umożliwia wyznaczenie dwóch rozłącznych krawędziowo ścieżek na podstawie własnego, zdefiniowanego grafu w pliku tekstowym o następującej strukturze:

```
## StartVertex EndVertex Weight
   a           b         1
   a           c         1
   a           e         2
   e           f         1
   f           d         1
```

gdzie komentarz rozpoczyna się od znaku #, każda linia reprezentuję pojedynczą krawędź rozpatrywanego grafu przy czym pierwsza kolumna opisuję etykietę wierzchołka początkowego danej krawędzi, kolumna druga definiuje etykietę wierzchołka końcowego, a w ostatniej kolumnie znajduje się waga konkretnej krawędzi. Wynik działania algorytmu będą reprezentowały listy etykiet wierzchołków najkrótszych ścieżek jeśli takie będą istniały oraz wagi konkretnych ścieżek.

4.2 Generowanie grafu

Funkcjonalność umożliwiająca podanie liczby wierzchołków oraz krawędzi oraz na tej podstawie wygenerowaniu przykładowego grafu. Następnie dla tak uzyskanego grafu następuje wyznaczenie dwóch najkrótszych ścieżek rozłącznych krawędziowo, wypisanie tych ścieżek na konsoli oraz wizualizacja graficzna takiego grafu.

4.3 Uruchomienie benchmarku

Ponadto aplikacja umożliwia dla wygenerowanego grafu na podstawie liczby wierzchołków oraz krawędzi wyznacznie ilości iteracji algorytmu potrzebnych do wyznaczenia rozłącznych ścieżek dla takiego przykładowo wygenerowanego grafu.

5 Kryteria stopu

Kryteriami stopu w rozpatrywanym algorytmie są osiągnięcie wierzchołka końcowego podanego na wejściu programu lub przejrzenie całej listy wierzchołków 'nieprzejrzanych' w kontekście wykonywanego zmodyfikowanego algorytmu Dijkstry.

6 Testy poprawności działania algorytmu

Poniżej zostały zamieszczone przykładowe wyniki wywołania programu dla zadanych wierzchołków w celu zbadania poprawności działania algorytmu.

6.1 Wejście programu

```
tartVertex EndVertex Weight
A B 5
A C 10
B D 20
B E 10
C D 18
C E 9
D E 10
```

6.2 Wyjście pgoramu

Program przedstawia dwie ścieżki sumarycznie najkrótsze i rozłączne krawędziowo oraz sumaryczną wagą ścieżki. Wyjściem programu dla przedstawionego wyżej wejścia jest:

6.3 Przypadek testowy 1

$A \quad - > B \quad (5)$
$A \quad - > C \quad (10)$
$A \quad - > D \quad (20)$
$B \quad - > D \quad (18)$
$C \quad - > E \quad (5)$
$C \quad - > I \quad (100)$
$D \quad - > E \quad (10)$
$D \quad - > F \quad (50)$
$D \quad - > H \quad (70)$
$E \quad - > F \quad (3)$
$E \quad - > G \quad (30)$
$E \quad - > H \quad (10)$
$F \quad - > G \quad (25)$
$F \quad - > H \quad (10)$
$G \quad - > I \quad (10)$
$H \quad - > I \quad (3)$

Oczekiwany wynik dla $A \quad - > I$:

- $A \quad - > C \quad - > E \quad - > H \quad - > I \quad (28)$
- $A \quad - > D \quad - > E \quad - > F \quad - > G \quad - > I \quad (68)$

6.4 Przypadek testowy 2

$A \quad - > B(1)$
$A \quad - > E(5)$
$A \quad - > F(2)$
$B \quad - > C(1)$
$B \quad - > E(2)$
$C \quad - > D(3)$

$E \quad -> D(3)$
$F \quad -> C(1)$
$F \quad -> D(6)$

Oczekiwany wynik:

- $A \quad -> \quad B-> E \quad -> D(6)$
- $A \quad -> F \quad -> C \quad -> D(6)$

6.5 Przypadek testowy 3

$A \quad -> B \quad (2)$
$A \quad -> C \quad (1)$
$A \quad -> I \quad (5)$
$B \quad -> C \quad (6)$
$C \quad -> D \quad (1)$
$C \quad -> E \quad (1)$
$D \quad -> E \quad (2)$
$D \quad -> F \quad (2)$
$E \quad -> F \quad (1)$
$E \quad -> K \quad (3)$
$F \quad -> G \quad (1)$
$F \quad -> I \quad (1)$
$F \quad -> J \quad (1)$
$F \quad -> K \quad (1)$
$G \quad -> H \quad (1)$
$I \quad -> E \quad (1)$
$I \quad -> J \quad (5)$
$J \quad -> H \quad (1)$
$K \quad -> H \quad (5)$

Oczekiwany wynik dla $A \quad -> H$:

- $A \quad -> C \quad -> D \quad -> F \quad -> G \quad -> H \quad (6)$
- $A \quad -> I \quad -> E \quad -> F \quad -> J \quad -> H \quad (9)$

7 Sytuacje awaryjne

W przypadku gdy nie istnieją dwie, rozłączne krawędziowo ścieżki pomiędzy wybranymi wierzchołkami, program zwróci odpowiedni komunikat informujący o tym użytkownika.

8 Testy wydajnościowe

Tabela 1 przedstawia czas wykonania zaimplementowanego algorytmu w zależności od wielkości grafu. Wiersze reprezentują ilość wierzchołków ze skokiem

$V_i = V_{i-1}^{1.05}$ oraz liczba krawędzi z mnożnikiem $E_i = 1.3 * E_{i-1}$. Wartości zostały dobrane arbitralnie na potrzeby uzyskania reprezentatwnych wyników. Na rysunku w każdej komórce występuje wartość mediany 10 wykonań algorytmu. Mediana pozwala na uzyskanie bardziej realnych wyników w porównaniu ze średnią z uwagi na mniejszą wrażliwość na odchylenia występujące w próbach.

9 Analiza złożoności obliczeniowej

Złożoność obliczeniowa algorytmu Dijkstry wynosi $O(E * \ln V)$, w naszym przypadku założylismy, że jego zmodyfikowana wersja będzie miała podobną złożoność. W obu poniższych tabelach główne wartości reprezentuje wartość mediany.

Rysunek 2 przedstawia analizę czasu wykonania algorytmu pod względem ilości krawędzi w grafie. W celu zbadania poprawności tej analizy potrzeba sprawdzić zgodność podanej proporcji pomiędzy wykonaniami dla kolejnych ilości krawędzie, a wartością oczekiwaną równą 1.3, która reprezentuje oczekiwany przelicznik złożoności w miarę proporcjonalnego zwiększania ilości krawędzi. Jak widac mediana tego przelicznika wynosi wartość 1.283 co jest bardzo zbliżoną wartością do wartości oczekiwanej.

Rysunek 3 przedstawia analizę czasu wykonania algorytmu pod względem ilości wierzchołków w grafie. W tym przypadku badane są proporcje odpowiednich logarytmów naturalnych. Wartością oczekiwaną w tym przypadku jest 1.05, natomiast wartością mediany uzyskaną na podstawie próbek jest 0.959, co jest wartością lekko rozbieżną. Jednakże, po analizie wartości mediany widać, że jej wartość wzrasta wraz z ilością wierzchołków i taka rozbieżność może wynikać ze zbyt małych prób wykorzystanych w tej analizie.

	Edge count																
Vertex count	**227**	**295**	**383**	**497**	**646**	**839**	**1090**	**1417**	**1842**	**2394**	**3112**	**4045**	**5258**	**6835**	**8885**	**11550**	**15015**	**19519**
100	9,65	8,11	10,20	12,89	16,26	22,90	27,97	37,25	75,57	102,87	204,85							
125	10,72	14,66	10,55	15,01	16,18	31,74	28,05	56,54	86,25	99,98	236,07	372,45	702,25					
159	21,38	22,50	21,39	15,15	26,87	37,09	46,24	59,06	99,30	107,01	281,00	394,06	755,98	932,46	1 914,93			
204	31,36	34,39	31,78	39,10	41,68	39,78	61,98	78,79	102,25	166,62	300,71	412,33	801,15	1 371,59	2 310,13	4 835,49	6 855,68	
266		34,74	43,53	49,57	67,65	79,77	91,72	111,10	121,62	231,04	274,01	480,06	810,88	1 291,31	2 256,37	2 338,93	5 790,11	11 408,01
351			83,68	81,50	88,02	92,32	128,61	175,54	146,71	167,58	321,96	615,05	575,02	1 338,00	2 375,78	3 906,10	7 497,30	8 137,20
470				158,99	162,06	200,22	205,57	197,34	268,47	359,72	342,01	528,32	891,16	1 098,45	2 319,57	2 641,11	6 867,11	12 066,61
639					371,26	279,02	218,07	203,67	383,07	425,03	476,00	541,67	1 105,13	1 625,78	2 284,09	4 227,34	5 886,47	10 482,26
882							428,92	564,56	605,13	611,26	725,30	1 034,66	826,89	1 848,94	1 983,15	3 908,33	3 350,05	12 257,32
1238								786,01	1 012,63	1 042,98	1 410,17	1 800,98	2 010,21	2 359,75	3 846,27	5 958,82	8 266,40	5 981,33
1767									2 195,09	2 566,64	1 652,93	2 475,38	3 923,91	4 689,18	4 717,92	5 913,32	6 484,66	7 657,49
2568											5 224,31	4 719,32	5 658,67	6 391,90	8 200,74	11 327,83	8 521,79	10 005,92
3802												9 539,21	10 781,39	4 894,36	7 660,61	11 827,04		

Rysunek 1: Czas wykonania.

$$f(v_k, e_l) = \frac{t(v_k, e_l)}{t(v_k, e_{l-1})}$$

stdev:	0,454
average:	1,393
median:	1,283
expected:	1,300

stdev:	0,217	0,232	0,251	0,268	0,364	0,257	0,311	0,375	0,277	0,623	0,276	0,434	0,529	0,359	0,353	0,660	0,909
average:	1,089	1,021	1,123	1,235	1,240	1,158	1,298	1,407	1,266	1,559	1,436	1,494	1,432	1,576	1,546	1,544	1,666
median:	1,074	0,950	1,184	1,080	1,207	1,186	1,277	1,329	1,159	1,352	1,427	1,636	1,233	1,630	1,547	1,392	1,469

stdev:	average:	median:
29,732	1,397	1,298
34,257	1,497	1,474
40,681	1,456	1,329
49,129	1,441	1,394
64,156	1,484	1,309
87,395	1,424	1,365
120,986	1,426	1,288
170,417	1,358	1,392
254,177	1,510	1,187
372,893	1,253	1,283
558,401	1,181	1,181
907,530	1,118	1,174
1 699,781	1,173	1,337

	295	**383**	**497**	**646**	**839**	**1090**	**1417**	**1842**	**2394**	**3112**	**4045**	**5258**	**6835**	**8885**	**11550**	**15015**	**19519**
100	0,840	1,257	1,264	1,262	1,408	1,221	1,332	2,029	1,361	1,991							
125	1,368	0,719	1,423	1,078	1,961	0,884	2,016	1,525	1,159	2,361	1,578	1,886					
159	1,052	0,950	0,709	1,773	1,380	1,247	1,277	1,681	1,078	2,626	1,402	1,918	1,233	2,054			
204	1,096	0,924	1,230	1,066	0,955	1,558	1,271	1,298	1,630	1,805	1,371	1,943	1,712	1,684	2,093	1,418	
266		1,253	1,139	1,365	1,179	1,150	1,211	1,095	1,900	1,186	1,752	1,689	1,592	1,747	1,037	2,476	1,970
351			0,974	1,080	1,049	1,393	1,365	0,836	1,142	1,921	1,910	0,935	2,327	1,776	1,644	1,919	1,085
470				1,019	1,235	1,027	0,960	1,360	1,340	0,951	1,545	1,687	1,233	2,112	1,139	2,600	1,757
639					0,752	0,782	0,934	1,881	1,110	1,120	1,138	2,040	1,471	1,405	1,851	1,392	1,781
882							1,316	1,072	1,010	1,187	1,427	0,799	2,236	1,073	1,971	0,857	3,659
1238								1,288	1,030	1,352	1,277	1,116	1,174	1,630	1,549	1,387	0,724
1767									1,169	0,644	1,498	1,585	1,195	1,006	1,253	1,097	1,181
2568											0,903	1,199	1,130	1,283	1,381	0,752	1,174
3802												1,130	0,454	1,565	1,544		

Rysunek 2: Analiza ze względu na ilość krawędzi.

$$f(v_k, e_l) = \frac{\ln(t(v_k, e_l))}{\ln(t(v_{k-1}, e_l))}$$

stdev:	0,061
average:	0,951
median:	0,959
expected:	1,050

stdev:	0,092	0,090	0,080	0,088	0,054	0,056	0,045	0,050	0,033	0,058	0,042	0,038	0,052	0,037	0,032	0,056	0,051	0,045
average:	0,873	0,880	0,882	0,896	0,899	0,921	0,929	0,936	0,945	0,950	0,959	0,962	0,970	0,979	0,984	0,990	0,998	1,003
median:	0,889	0,871	0,885	0,913	0,882	0,941	0,921	0,934	0,951	0,942	0,972	0,961	0,969	0,976	0,998	0,995	1,014	0,983

stdev:	average:	median:
0,067	0,947	0,970
0,085	0,921	0,969
0,068	0,928	0,934
0,062	0,965	0,970
0,057	0,965	0,965
0,067	0,950	0,958
0,042	0,967	0,971
0,068	0,962	0,963
0,053	0,940	0,926
0,046	0,954	0,966
0,035	0,940	0,948
0,048	0,978	0,995

	227	**295**	**383**	**497**	**646**	**839**	**1090**	**1417**	**1842**	**2394**	**3112**	**4045**	**5258**	**6835**	**8885**	**11550**	**15015**	**19519**
125	0,956	0,779	0,986	0,944	1,002	0,906	0,999	0,897	0,970	1,006	0,974							
159	0,775	0,862	0,769	0,996	0,846	0,957	0,870	0,989	0,969	0,985	0,969	0,991	0,989					
204	0,889	0,880	0,885	0,741	0,882	0,981	0,929	0,934	0,994	0,913	0,988	0,992	0,991	0,947	0,976			
266		0,997	0,917	0,939	0,885	0,841	0,913	0,927	0,964	0,940	1,017	0,975	0,998	1,008	1,003	1,094	1,019	
351			0,852	0,887	0,941	0,968	0,930	0,911	0,962	1,063	0,972	0,961	1,054	0,995	0,993	0,938	0,971	1,038
470				0,868	0,880	0,854	0,912	0,978	0,892	0,870	0,990	1,024	0,935	1,028	1,003	1,050	1,010	0,958
639					0,860	0,941	0,989	0,994	0,940	0,972	0,946	0,996	0,969	0,947	1,002	0,944	1,018	1,015
882							0,888	0,839	0,929	0,943	0,936	0,907	1,043	0,983	1,019	1,009	1,069	0,983
1238								0,950	0,926	0,923	0,908	0,926	0,883	0,969	0,920	0,951	0,900	1,083
1767									0,899	0,885	0,979	0,959	0,919	0,919	0,976	1,001	1,028	0,972
2568											0,866	0,924	0,958	0,965	0,939	0,930	0,970	0,971
3802												0,923	0,931	1,031	1,008	0,995		

Rysunek 3: Analiza ze względu na ilość wierzchołków.