

Teoria dos Grafos – COS 242

Bruno Dantas de Paiva, Eduardo Guedes de Seixas

Escola Politécnica – Universidade Federal do Rio de Janeiro (UFRJ)

Rio de Janeiro – RJ – Brasil

Introdução

A ideia principal do trabalho é desenvolver uma biblioteca para manipular grafos, que seja capaz de agregar também algoritmos essenciais para o trabalho.

Implementação

A implementação ocorreu na linguagem C++, onde optamos por um conjunto de funções. Contudo, foi necessária a criação de duas `structs` para auxiliar a utilização dos dados que representavam os grafos nos algoritmos implementados. Elas são:

```
grafoVector
int numVertices;
vector<pair<int,double> >*adjVector;
```

```
grafoMatriz
int numVertices;
double **adjMatriz;
```

Imagem 1: Estruturas criadas para facilitar a integração do grafo com o código.

a- Principais observações sobre o projeto:

1- Adjunto ao trabalho passado, foram implementadas outras duas `structs` a fim de facilitar a integração do usuário. Deste modo, as `structs` já implementadas foram reimplementadas com pequenas alterações, enquanto a Matriz de Adjacência agora está implementada com `double` e o Vetor de Adjacência está implementado com vetor de `pair`, onde `pair` é uma estrutura de dados implementada em c++ cujos dados são inseridos em forma de pares, onde o primeiro elemento do par é o vértice vizinho e o segundo elemento do par o peso entre eles.

2- Durante a execução da biblioteca, foi utilizada a flag `-O3` para otimização do código, pois, por testes de caso, pode-se observar que esta gerou um desempenho melhor no tempo de execução, diminuindo-o.

3- Ao analisar os estudos de caso, para o grafo rede de colaboração, foi observado que o grafo não era conexo, ou seja, o algoritmo de mst não engloba tal tipo de grafo (este iria compor uma `msf` – minimum spanning forest), contudo, para fins didáticos, os resultados foram obtidos rodando uma mst a partir de Edsger W. Dijkstra. Tal relação também vale para a excentricidade (não engloba grafos desconexos).

4- Atentando aos detalhes durante a apresentação do trabalho passado, foram acopladas a algumas funções uma flag booleana (`salve`) que permite o usuário escolher salvar ou não um arquivo `.txt` contendo a informação referente àquela função.

5- Nota-se que a implementação de dijkstra e prim não foi realizada da melhor maneira (heap binária) e sim com set de pair, deste modo, os tempos encontrados nas tabelas para a Excentricidade e Prim não são ótimos. Tal fato é considerado uma decisão de projeto que deve ser modificada futuramente.

6- Todo o trabalho e implementações podem ser encontrados no seguinte repositório: <https://github.com/DantasB/Graph-Library>

b- Estudo de Caso:

Grafo Vértice	10	20	30	40	50	Tempo médio (s)
1	44	55	61	42	48	0,00050391
2	54	55	53	50	52	0,0104857
3	58	58	64	57	60	0,258011
4	1650	1721	1628	1700	1671	0,743275
5	35482639	35482639	35482639	35482639	35482639	2,43904
Rede de Colaboração	116,2484 138524	115,20962 38245	115,17135 92927	116,4779 733282	115,62230 07658	1,09919

Tabela 1: Estudos de casos da excentricidade para os grafos requisitados..

Grafo	Peso	Tempo (s)
1	3856	0,000973
2	31663	0,013793
3	312289	0,293072
4	7668215	0,363579
5	79992951	2,20004
Rede de Colaboração (Origem:Edoardo Airoidi)	1274196, 38070883	0,950563

Tabela 2: Estudos de casos da MST para os grafos requisitados.

Grafo	10	20	30	40	50
1Dist.*	31	38	48	25	30
Path	[10, 268, 484, 798, 133, 710, 709, 881, 1000, 1]	[20, 21, 416, 141, 604, 4, 3, 2, 1]	[30, 29, 28, 134, 74, 709, 881, 1000, 1]	[40, 74, 709, 881, 1000, 1]	[50, 51, 919, 768, 881, 1000, 1]

2 Dist.	9	10	9	8	7
Path	[10, 8767, 9110, 8275, 5675, 1]	[20, 21, 979, 4992, 670, 3985, 1]	[30, 5021, 5020, 9823, 2538, 3985, 1]	[40, 6804, 6758, 8166, 8167, 5247, 2494, 1]	[50, 7672, 857, 3985, 1]
3 Dist.	5	10	16	8	7
Path	[10, 99499, 93705, 52193, 1]	[20, 21, 95822, 10027, 95946, 27287, 9824, 43169, 43170, 1]	[30, 31, 60660, 44171, 95621, 82358, 15899, 1]	[40, 27630, 77071, 43169, 43170, 1]	[50, 21939, 93705, 52193, 1]
4 Dist.	58	129	102	174	147
Path	[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]	[20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]	[30, 31, 19296, 627814, 627813, 627812, 723121, 723120, 103161, 103162, 103163, 126309, 164224, 387113, 556565, 15847, 101894, 814196, 814195, 396781, 396782, 5, 4, 3, 2, 1]	[40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 19296, 627814, 627813, 627812, 723121, 723120, 103161, 103162, 103163, 126309, 164224, 387113, 556565, 15847, 101894, 814196, 814195, 396781, 396782, 5, 4, 3, 2, 1]	[50, 51, 52, 53, 54, 55, 390888, 286933, 906093, 285753, 285752, 285751, 285750, 285749, 285748, 285747, 929180, 596549, 811148, 811147, 811146, 5, 4, 3, 2, 1]
5Dist.	93	150	241	332	400
Path	[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]	[20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]	[30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]	[40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]	[50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
Colabo ração Dist.	2,8065476 1904762	2,379356238832 44	2,683963270362 95	3,179197994987 47	2,420577123874 28
Path	[10, 2574, 83518, 258146, 7783, 3244, 1]	[20, 3984, 353044, 5161, 9971, 2524, 4586, 12822, 537589, 3244, 1]	[30, 4106, 220549, 688962, 209457, 7314, 5306, 3244, 1]	[40, 5265, 272393, 3244, 1]	[50, 556113, 650795, 12573, 13155, 13071, 6885, 426314, 3244, 1]

Tabela 3: Estudos de casos de distâncias e caminhos para os Grafos requisitados.

*Dist. = Distância medida durante o algoritmo de Dijkstra.

Distância	Alan M. Turing	J. B. Kruskal	Jon M. Kleinberg	Eva Tardos	Daniel R. Figueiredo
Edsger W. Dijkstra	Infinito	3.480368	2.706993618	2.753514179	2.94283087
Caminho Mínimo	[]	[J. B. Kruskal, Albert G. Greenberg, R. Srikant, Ness B. Shroff, Edwin K. P. Chong, Howard Jay Siegel, Dan C. Marinescu, John R. Rice, Edsger W. Dijkstra]	[Jon M. Kleinberg, Prabhakar Raghavan, Eli Upfal, Avi Wigderson, Prabhakar Ragde, Dimitrios M. Thilikos, Hans L. Bodlaender, Gerard Tel, A. J. M. van Gasteren, Edsger W. Dijkstra]	[Éva Tardos, Serge A. Plotkin, Andrew V. Goldberg, Robert Endre Tarjan, Haim Kaplan, Micha Sharir, Mark H. Overmars, Jan van Leeuwen, Hans L. Bodlaender, Gerard Tel, A. J. M. van Gasteren, Edsger W. Dijkstra]	[Daniel R. Figueiredo, Donald F. Towsley, Zhi-Li Zhang, Y. Thomas Hou, Bo Li, Chuang Lin, Dan C. Marinescu, John R. Rice, Edsger W. Dijkstra]

Tabela 5: Estudos de casos da rede de colaboração a partir de Edsger W. Dijkstra.

Maiores Graus na MST	Grau
Wei Li	171
Wei Wang	146
Wei Zhang	143
Vértice	Vizinhos na MST
Daniel R. Figueiredo	[Alexandre A. Santos, André C. Pinho, Donald F. Towsley]
Edsger W. Dijkstra	[A. J. M. van Gasteren, Carel S. Scholten, W. Heise]

Tabela 6: Estudos de casos da rede de colaboração sobre maiores graus e vizinhos.

C – Considerações:

1- Devido os algoritmos terem sido implementados como conjunto de funções, algumas funcionalidades (tal como retornar o caminho mínimo) foram implementadas separadamente nas funções dijkstra e bfs, tratando assim a questão do grafo ter ou não peso nas arestas.

2- É importante ressaltar que foi criado um arquivo externo para o tratamento dos dados da rede de colaboradores, justamente por questão de organização. Contudo, tal arquivo será integrado posteriormente a biblioteca.