



UNIVERSIDADE ESTADUAL DE MARINGÁ

DEPARTAMENTO DE INFORMÁTICA

6898 – ALGORITMOS EM GRAFO

Professor: Marco A. L. Barbosa

RELATÓRIO

IMPLEMENTAÇÃO E TESTE DE ALGORITMOS DE GERAÇÃO DE ÁRVORES ALEATÓRIAS

GABRIEL THIAGO HENRIQUE DOS SANTOS

RA: 107774

MARINGÁ

2019

O trabalho com a implementação de maneira geral foi proveitoso, tendo as entregas semanais das funções correspondente fundamentais para um código completo ao final, sendo útil não apenas no fim, mas sim no decorrer, no meu caso, pelo menos, aprofundando em uma linguagem (Python) em que não tinha muita experiência, também sendo conveniente pois não houve um aperto para última hora como se pode acontecer em alguns casos quando chega em momentos de provas e trabalhos paralelos a outras disciplinas. Ocorreu algumas dificuldades/erros no decorrer, no entanto, foi-se superado muitas vezes por pesquisas pela internet ou quando não atendidas claramente, um encaminhamento a sala do professor para um solução da dúvida ou esclarecimento sendo totalmente benéfico, ao meu ver, dificuldades de entendimento de algumas partes do trabalho foi a de não bater com o decorrer da aula, ou seja, em momentos uma certa função sendo ter que começar a implementação antes da explicação do mesmo, entretanto em próximas aulas foram mais claras sobre tais, ou até mesmo, voltando a uma conversa antecipada sobre a implementação antes de aula.

Foi feito então três algoritmos para geração de árvores aleatórias: o primeiro baseado em passeio aleatório (Random Walk), em seguida o algoritmo de Prim e o último o algoritmo de Kruskal.

Veremos as tabelas para comparação dos tempos de execução médio de cada algoritmo para cada valor de n , os tempo :

Algoritmo baseado em passeio aleatório (Random Walk)

valor de n	diâmetro médio	tempo (seg)
250	46.752	3.18
500	70.044	6.98
750	86.196	11.67
1000	101.568	15.29
1250	112.51	21.11
1500	123.872	24.77
1750	133.83	29.00
2000	143.478	33.03
TOTAL		145.03 (ou 2.4 min)

Algoritmo de Kruskal

valor de n	diâmetro médio	tempo (seg)
250	37.116	61.43
500	50.426	278.73
750	59.746	675.60
1000	66.87	1294.30
1250	73.85	2101.02
1500	78.632	3088.19
1750	83.734	4103.13
2000	89.24	5455.14
TOTAL		17057.54 (ou 4.5 hr)

Algoritmo de Prim

valor de n	diâmetro médio	tempo (seg)
250	36.79	29.36
500	49.864	134.63
750	59.58	333.11
1000	66.998	654.14
1250	73.556	1079.96
1500	80.004	1594.89
1750	83.8	2238.70
2000	89.09	2952.48
TOTAL		9017.27 (ou 2.5 hr)

Como o esperado, o algoritmo Random walk executou para todos os valores de n , em tempo menor, já que como o nome mesmo diz, percorre um caminho aleatório, ou seja, a escolha não ocorre com tais verificações como os outros dois algoritmos em seguida. O segundo tempo menor foi o de Prim, que como já estudados ele poderia dependendo da implementação ter um tempo significativo quando comparado ao de Kruskal, e este então se ocorreu de maneira justa. Por último, o de Kruskal apesar de ter ficado em último, a complexidade não foi de maneira esperada totalmente, a implementação não se sucedeu com uso de classes por exemplo em que não foi possível, logo então na medida possível foi o mesmo solucionado, porém por um caminho maior, ocasionando em um tempo além do mesmo. No entanto, de maneira geral os tempos e implementações foi feita de maneiras claras, com resultados esperados e com a complexidade seguras para a correção.

Em seguida, mostraremos os gráficos com os resultados dos diâmetros médios de cada algoritmo:



