

Lista 3 - MAC0328

Matheus de Mello Santos Oliveira - 8642821

1 Experiência

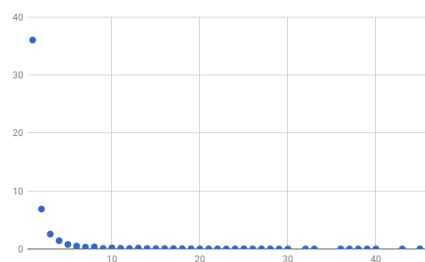
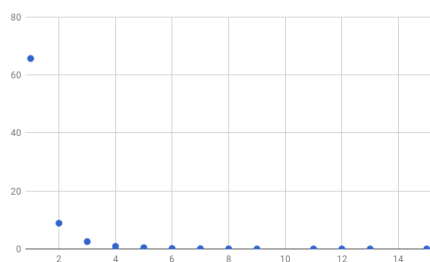
O programa segue o experimento descrito na questão proposta.

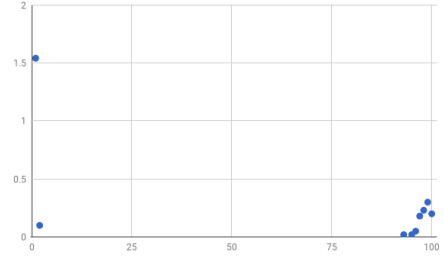
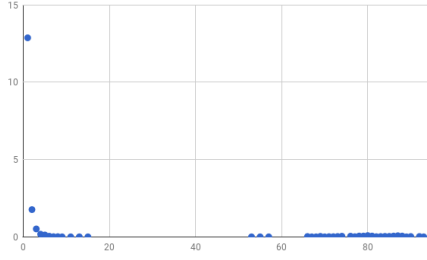
1. Seja G um grafo não dirigido aleatório com V vértices e quantidade de arestas esperadas E , $E = V * \{0.2, 0.5, 1, 2, 5, 10, 20\}$.
2. DFS para calcular as componentes conexas e seus respectivos tamanhos.
3. Guarda a frequência de cada tamanho.
4. Repete itens 1-3 T vezes.
5. Repete itens 1-4 para cada quantidade de aresta.
6. Imprime uma tabela com a média das frequências dos tamanhos das componentes conexas após T testes e E arestas esperadas.

2 Resultados

Primeiramente é preciso notar que é esperado que o aumento na quantidade de testes nos traga um refinamento da informação adquirida, pois ao aumentar o número de testes, diminuimos os riscos de casos atípicos contaminarem o resultado, estabilizando tanto média, variância quanto desvio padrão. Com isto em mente a quantidade de testes padrão foi estabelecida como 100 para o desenvolvimento desse projeto. Como para cada valor de V temos 7 variações da quantidade de arestas serão exibidos os resultados para $V = 100$ e $V = 1000$.

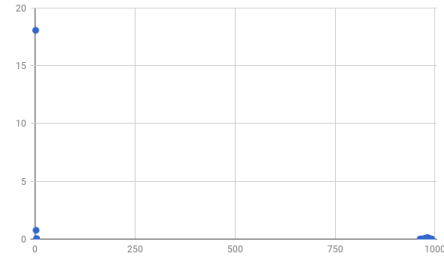
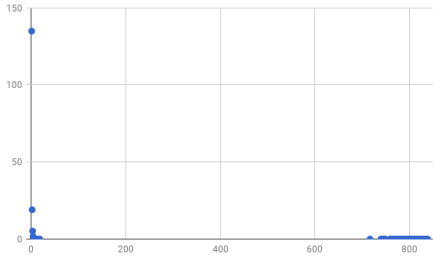
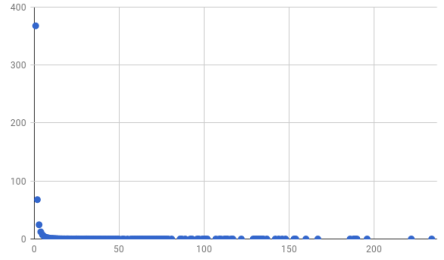
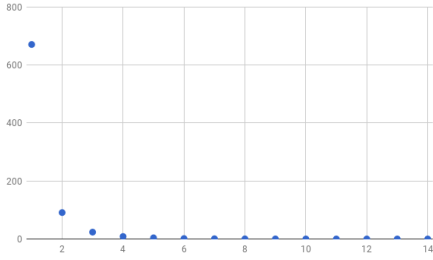
$V = 100$





As tabelas para 500, 1000 e 2000 arestas se resumem a apenas 1 componente com 100 vértices. Isto de certa forma confirma o que queríamos comprovar empiricamente, que a partir de $V \log_2(V) = 100(6.64)/2 = 332$, e além disso percebemos que para 100 arestas, a maioria dos testes consiste de componentes com em torno de 60 vértices e várias outras componentes satélites com poucos vértices cada.

$V = 1000$



As tabelas para 5000, 10000 e 20000 arestas se resumem a apenas 1 componente com 1000 vértices. Isto claramente confirma o que queríamos comprovar empiricamente, que a partir de $V \log_2(V) = 1000(9.96)/2 = 4980$ teríamos um grande componente com outras pequenas como satélites, mas neste caso percebemos que a partir de 1000 arestas .

3 Implementação

Este código foi implementado utilizando matriz de adjacência principalmente pelo fato de que o gargalo do desempenho está na própria criação do grafo, que tem consumo de tempo: $O(V^2)$. Não

somente isto, matriz de adjacência permite inserção de aresta em tempo constante, ao passo que listas de adjacência tem pior caso linear no número de vértices o que tornaria a criação de grafos aleatórios $O(V^3)$

4 Uso do programa

Para compilar:

```
$ make
```

Para executar o código:

```
$ ./ep3 [V] [T]
```

Onde V é o número de vértices e T o número de testes a serem executados para cada quantidade de arestas.