

Algoritmos e Estruturas de Dados 2 - EP2

Lourenço Henrique Moinheiro Martins Sborz Bogo

1 Escolhas de Implementação

Primeiro problema que encontrei ao começar a implementar o EP foi: como eu saberia em que vértice uma certa palavra está, de maneira rápida? Para resolver isso, eu decidi usar uma HashTable, onde as chaves são as strings que contêm as palavras e os valores são os nodos nos quais elas estão.

Para implementar a HashTable, decidi fazer minha própria biblioteca de lista ligada, que transformei em uma biblioteca de fila para que pudesse ser usada na BFS também.

Depois disso, implementar quase todas as funções do EP foi muito simples, com exceção das `emCiclo`, que demorei um pouco para pensar como fazer. Decidi implementar quase tudo usando DFS, menos o cálculo da distância onde optei por usar BFS.

Tive que fazer duas funções DFS auxiliares:

`int dfs(int)` Roda uma dfs para o nodo passado como parâmetro e devolve o tamanho de sua componente.

`void dfs(int, int, int, bool, int)` Essa dfs foi a parte mais complicada de implementar do EP. Caso sejam passados só os 3 primeiros parâmetros para ela, ela irá procurar se existe ou não um ciclo que contém o primeiro parâmetro. Caso os 5 parâmetros sejam passados, a função irá fazer quase a mesma coisa, com a condição de que ao invés de buscar um ciclo no primeiro parâmetro, ela irá buscar um ciclo no primeiro parâmetro, que contenha o último parâmetro.

Além dessas funções, implementei mais algumas auxiliares para descobrir a maior componente, a menor componente e o tamanho médio da componente, que são calculadas junto com o número de componentes, ou seja, quando um desses dados é pedido, todos são calculados para ganhar tempo.

Para achar a distância média entre vértices, preferi não usar a função que acha a distância entre dois vértices, pois ficaria muito lerdo. Preferi fazer uma função separada que roda uma BFS para cada nodo e depois tira a média de todas as distâncias encontradas.

2 Como usar

Para usar o EP, deve-se passar como argumento na linha de comando o `k` descrito no enunciado.

Depois disso, basta digitar `help` na prompt, e aparecerá explicações para todos os comandos possíveis.

3 Experimentos

Vamos testar como o tipo do texto, sua língua e seu tamanho influenciam os resultados.

Livro	Vertices	Arestas	Componentes	Maior Comp.	Menor Comp.	Média Comp.	Dist. Média	Denso
Memórias	10976	8293	6307	2715	1	1.738	8.71	False
Dom Casmurro	9475	7825	5115	2393	1	1.850	8.489	False
Quincas	11335	9486	6195	2829	1	1.820	8.740	False
Huckleberry	7301	8218	3647	2647	1	1.999	7.237	False
The Raven	445	197	309	51	1	1.407	4.393	False
Discourse	2819	1151	2040	372	1	1.412	7.520	False
Critique	7995	4593	5461	952	1	1.462	7.227	False
Manual GCC	11781	11984	7083	2267	1	1.661	6.543	False
Manual GHC	1702	797	1307	131	1	1.294	4.380	False
Manual wget	2400	1591	1669	374	1	1.432	6.544	False
Manual grep	962	860	672	132	1	1.416	4.107	False

OBS Todos os testes feitos na tabela a cima foram feitos com $k = 1$;

Percebemos com os testes feitos acima (e mais alguns q não coloquei na tabela) que, os fatores que mais influenciam os resultados são o tamanho do texto e o seu gênero, o que faz bastante sentido.

O tamanho faz uma diferença gritante pois, quanto mais palavras têm um texto, mais difícil fica de usar palavras diferentes e mais palavra se conectam, fazendo com que o número de arestas fique mais próximo ou até maior que o número de vértices em alguns casos. Já, nos textos menores, as palavras têm menos oportunidades de se conectar com as outras, fazendo com que o número de arestas fique bem menor que o número de vértices.

O gênero influencia menos que o tamanho no resultado, mas, podemos perceber que a média das componentes é menor nos manuais, do que nos livros de romance.

Outra coisa que se nota é que nenhum dos grafos é denso, o que também é de certa maneira óbvio já que a condição para um grafo ser denso muito provavelmente não será atendida por nenhum texto real, com as condições que o EP impôs.

A diferença entre as línguas só se torna evidente quando variamos o k . Por exemplo, textos em alemão tendem a se comportar de maneira diferente dos outros com um k maior, pois as palavras tendem a ser maiores. Em português o k ser alterado de 1 pra 3 faz muita diferença por causa dos artigos e preposições de 1 ou 2 letras que temos na língua.