



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”

Câmpus de Rio Claro

Relatório de estudo sobre grafos do tipo árvore

Grafos e Aplicações

Equipe:

André Luis Dias Nogueira
Felipe Melchior de Britto
Rafael Daiki Kaneko
Ryan Hideki Tadeo Guimarães
Vitor Marchini Rolisola

01/08/2024

Conteúdo

1	Resumo	3
2	Introdução	4
3	Implementação	7
4	Resultados e discussão	8
5	Conclusão	9

1 Resumo

Este trabalho tem como objetivo aplicar o algoritmo de Busca em Largura (BFS) em árvores binomiais geradas aleatoriamente, calculando a média das profundidades das ordenações geradas para todos os nós. O número de filhos por nó segue uma distribuição binomial com parâmetros n_{\max} (número máximo de filhos) e p (probabilidade de ter filhos), sendo a geração de descendentes limitada pela profundidade d_{\max} .

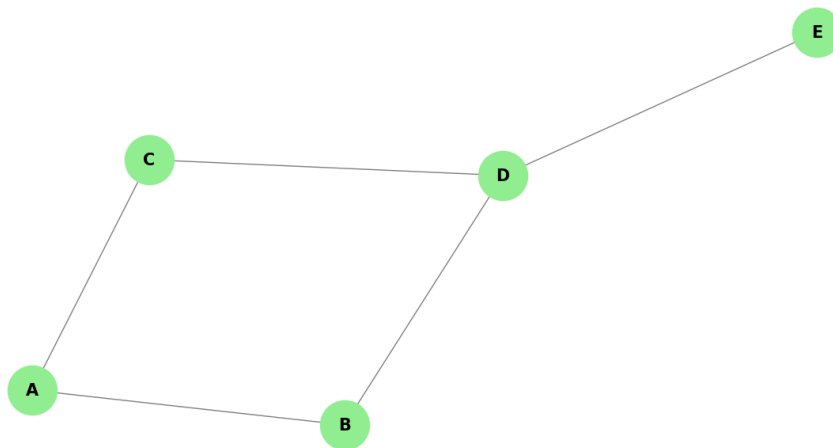
Realizamos experimentos variando os parâmetros n_{\max} , p , e d_{\max} , gerando ao menos 10 árvores aleatórias para cada combinação de valores. Os resultados obtidos foram apresentados em tabelas e gráficos, permitindo a análise das profundidades médias.

Um dos interesses do estudo era almejar a transição de fase esperada no modelo: para $n_{\max} \cdot p < 1$, as árvores atingem no geral níveis sem descendentes enquanto para $n_{\max} \cdot p > 1$, as árvores se expandem até o limite de profundidade d_{\max} . Esse comportamento fornece uma visão mais detalhada sobre a estrutura de crescimento das árvores binomiais conforme variamos seus parâmetros.

2 Introdução

Grafos são estruturas fundamentais em teoria dos grafos, utilizadas para modelar uma variedade de problemas em diferentes áreas, desde redes de computadores até genética.

Um grafo é uma estrutura matemática usada para modelar relações entre objetos de um conjunto. Ele é composto por dois conjuntos: um conjunto de vértices (ou nós) e um conjunto de arestas (ou arcos) que conectam esses vértices. Os vértices representam os objetos e as arestas representam as relações entre esses objetos. ^[1]



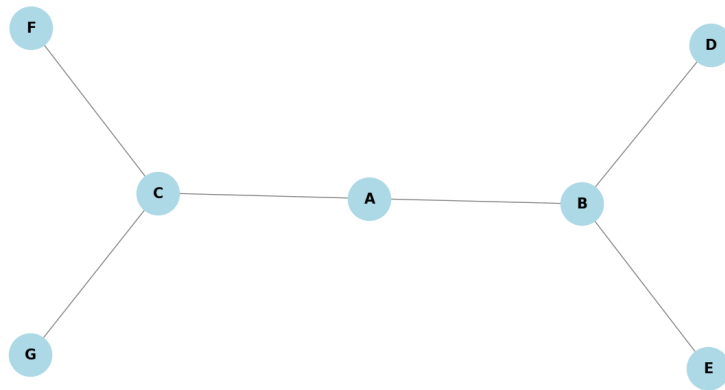
Por exemplo, na representação de uma rede social, os vértices podem representar pessoas e as arestas representam as conexões de amizade entre elas.

Um tipo especial de grafo, conhecido como árvore, apresenta propriedades únicas que tornam essa classe particularmente interessante para estudo.

Uma árvore é definida como um grafo não-orientado, conexo e acíclico, o que significa que não possui ciclos e, além disso, qualquer remoção de uma de suas arestas resulta em um grafo desconexo.^[4] Então suas características são:

- **Conectividade:** Para qualquer par de vértices u e v , existe exatamente um caminho que conecta u e v .
- **Aciclicidade:** O grafo não contém ciclos; ou seja, não é possível iniciar em um vértice, seguir arestas e retornar ao mesmo vértice sem atravessar arestas repetidamente.
- **Número de arestas:** Se uma árvore possui n vértices, então ela possui exatamente $n - 1$ arestas.

Essas características permitem que árvores sejam a estrutura mínima necessária para garantir a conectividade entre os vértices de um grafo com o menor número de arestas possíveis, um aspecto crucial para a otimização de recursos em diversos cenários práticos.

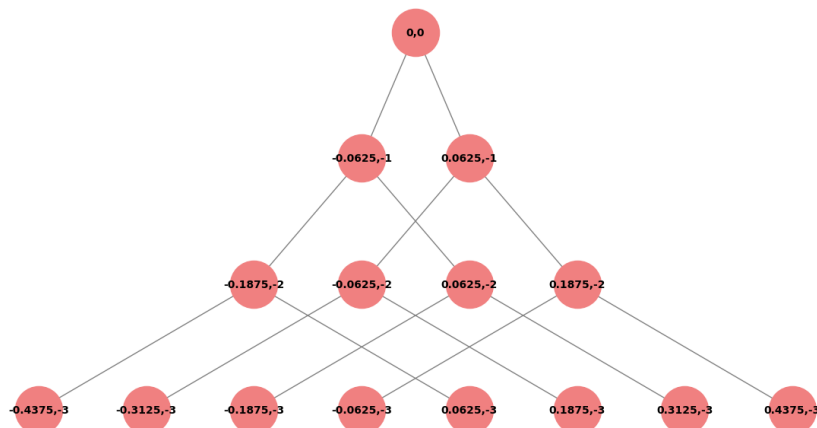


A análise de árvores em grafos tem implicações diretas em problemas de interligação, como o fornecimento de redes elétricas, onde o objetivo é minimizar o custo de conexão ao garantir que todas as unidades estejam conectadas. Além disso, árvores desempenham um papel importante na computação, particularmente em algoritmos de ordenação, como o Heapsort, e na modelagem de genealogias e redes hierárquicas.

Uma **árvore binomial** é uma estrutura de dados que representa uma coleção de árvores binomiais. A definição formal de uma árvore binomial é a seguinte^{[3] [2]}:

Uma **árvore binomial** B_k é uma árvore que possui as seguintes propriedades:

- **Estrutura Recursiva:** Uma árvore binomial B_k é composta por 2^k nós e tem exatamente k árvores binomiais $B_{k-1}, B_{k-2}, \dots, B_0$ como subárvores. A árvore B_k é obtida ao unir duas árvores B_{k-1} .
- **Propriedades dos Nós:**
 - O nó na raiz de B_k tem um grau de k (ou seja, ele possui k filhos).
 - A altura de B_k é k .
 - A árvore B_k possui 2^k folhas.
- **Organização dos Nós:** Os nós são organizados de tal forma que os valores dos nós na subárvore esquerda são menores ou iguais ao valor do nó pai, e os valores dos nós na subárvore direita são maiores.



As árvores binomiais são particularmente úteis em algoritmos de estrutura de dados, como em filas de prioridade.

O presente relatório tem o objetivo de explorar as propriedades matemáticas e aplicativas das árvores binomiais, abordando tanto sua definição formal quanto suas extensões, como arborescências e a aplicação em algoritmos de busca.

A introdução a essas ideias será contextualizada com base nas propriedades da conectividade e da aciclicidade, discutindo ainda como árvores binomiais podem ser vistas como estruturas mínimas e otimizadas para representação de relações complexas, ao mesmo tempo que mantêm a simplicidade computacional.

3 Implementação

A implementação do trabalho envolve a criação de um algoritmo que gera árvores binomiais com base nos parâmetros fornecidos: n_{\max} (número máximo de filhos por nó), p (probabilidade de um nó gerar filhos) e d_{\max} (profundidade máxima da árvore). Cada nó é representado por um conjunto de quatro posições em um vetor. A primeira posição do vetor, exclusivamente, armazena o número máximo de filhos do nó, enquanto as outras três posições armazenam o índice do pai e dos filhos, respectivamente, conforme o modelo que você está utilizando.

Para cada tripla de parâmetros, são geradas pelo menos 10 árvores aleatórias, utilizando a distribuição binomial para decidir quantos filhos cada nó terá. O algoritmo segue uma abordagem de Busca em Largura (BFS) para explorar os nós da árvore. A BFS percorre a árvore começando da raiz, explorando todos os nós em cada nível antes de passar para o próximo. Essa busca é gerenciada por uma fila que armazena os nós a serem explorados, garantindo que os nós sejam processados em ordem de nível.

Durante a execução do BFS, são registrados a profundidade de cada nó e a profundidade total das árvores. Ao final de cada execução, é calculada a profundidade média da árvore, que será utilizada para gerar tabelas ou gráficos que mostram como essa profundidade varia de acordo com as diferentes combinações de n_{\max} , p , e d_{\max} .

Além disso, o código precisa lidar com a transição de fase do modelo, que ocorre quando n_{\max} multiplicado por p é menor que 1, indicando que, em algum ponto, a geração de novos filhos para a árvore vai parar, resultando em níveis sem descendentes. Caso $n_{\max} \cdot p$ seja maior que 1, a geração de nós continuará até atingir o limite imposto por d_{\max} .

Portanto, a implementação é construída de maneira a não apenas gerar as árvores e calcular as profundidades, mas também simular a dinâmica de crescimento dessas árvores binomiais sob diferentes condições, explorando a variação desses parâmetros.

4 Resultados e discussão

5 Conclusão

Bibliografia

- [1] Emilio Bergamin Junior. *Aula 1 - Conceitos introdutórios de grafos*. 2024. URL: <https://drive.google.com/file/d/1ZE6hkZ3LWcdHdRhw44ctXAguC0Iqbqzm/view>.
- [2] Daniel D. Sleator Robert E. Tarjan. «A data structure for dynamic trees». Em: *STOC '81: Proceedings of the thirteenth annual ACM symposium on Theory of computing* (1981). DOI: 10.1145/800076.
- [3] Ronald L. Rivest e Clifford Stein Thomas H. Cormen Charles E. Leiserson. *Introduction to Algorithms*. First edition. MIT Press, 1990.
- [4] Douglas B. West. *Introduction to Graph Theory*. Second edition. Pearson College Div, 2000.