

Configuração do computador:

SO: Windows 10

Processador: Ryzen 7

Memória: 16 gb RAM DDR4 3200Mhz

Armazenamento: SSD M2 Nvme 1 tera

O que tem nos .csv:

vértice: O número de vértices no grafo.

tempo_execução: O tempo necessário para executar o algoritmo e encontrar uma solução (medido em segundos).

solução: O comprimento da solução encontrada pelo algoritmo, ou seja, o número de vértices visitados na rota encontrada.

rota: A sequência de vértices visitados na solução encontrada pelo algoritmo.

Análise dos resultados:

Podemos observar no algoritmo de força bruta que o tempo de execução é consideravelmente maior à medida que o tamanho do grafo aumenta. Por exemplo, para um grafo com 5 vértices, o tempo de execução é em torno de 0.001 segundos, enquanto para um grafo com 10 vértices, o tempo aumenta para aproximadamente 1.2 segundos. Essa tendência de aumento no tempo de execução mostra que o algoritmo de força bruta se torna menos eficiente à medida que o tamanho do problema aumenta.

No algoritmo de backtracking, podemos observar que o tempo de execução é significativamente menor em comparação ao algoritmo de força bruta para grafos maiores. Por exemplo, para um grafo com 10 vértices, o tempo de execução varia de cerca de 0.1 a 0.5 segundos. Isso demonstra que o algoritmo de backtracking é mais eficiente para problemas maiores, pois consegue encontrar soluções viáveis em um tempo razoável.

Analisando os resultados da programação dinâmica, podemos observar que o tempo de execução é muito menor em comparação aos algoritmos de força bruta e backtracking, mesmo para grafos maiores. Por exemplo, para um grafo com 10 vértices, o tempo de execução é em torno de 0.001 segundos. Essa eficiência na execução se deve à natureza recursiva e à estratégia de armazenamento de subproblemas resolvidos, que evita o retrabalho desnecessário.

Comparando os resultados com a primeira parte:

Na primeira parte foi construída essa tabela:

N de vértices	Tempo médio de execução Força bruta	Tempo de médio de execução Guloso	Contagem de soluções iguais
5	0.14 ms	0.02 ms	350
6	0.8 ms	0.02 ms	237
7	6.49 ms	0.02 ms	144
8	58.38 ms	0.04 ms	101
9	575.01 ms	0.06 ms	73

Observando os valores, podemos notar que o algoritmo Guloso possui o menor tempo médio de execução, mantendo-se em torno de 0.02 ms independentemente do número de vértices. Isso sugere uma eficiência superior em relação aos outros algoritmos.

Por outro lado, a abordagem de Programação Dinâmica apresenta tempos de execução um pouco maiores do que o Guloso, mas ainda muito pequenos (em torno de 0.07 ms para 5 vértices). No entanto, é importante frisar que o algoritmo foi testado em máquinas diferentes.

Já o algoritmo de Força Bruta demonstra um aumento significativo no tempo de execução à medida que o número de vértices aumenta, com tempos variando de 0.14 ms para 5 vértices até 575.01 ms para 9 vértices. Essa abordagem tem uma complexidade exponencial, o que a torna menos eficiente em problemas maiores.

O Backtracking por sua vez apresentou um tempo médio de execução menor do que o Força Bruta na maior parte dos casos e em relação ao Guloso, o Backtracking apresentou um tempo médio de execução de aproximadamente 0.124, enquanto o Guloso possui um tempo de execução de cerca de 0.02 ms.

Portanto, com base nos “.csv”, foi possível ver que o algoritmo Guloso parece ser o mais eficiente em termos de tempo de execução, seguido pela Programação Dinâmica. A Força Bruta é a abordagem menos eficiente em termos de tempo, especialmente para problemas com um número maior de vértices.

Conclusão:

A escolha de qual algoritmo usar no problema vai depender do tamanho do grafo, disponibilidade de tempo e a necessidade de uma solução ótima, o algoritmo Backtracking é mais indicado do que a força bruta desde que seu tempo de execução esteja dentro de limites aceitáveis. Em grafos maiores o indicado seria o algoritmo guloso, mas caso precise de um algoritmo garanta a solução ótima o indicado é a programação dinâmica, que possui boa eficiência em termos de tempo de execução. O algoritmo guloso encontra soluções aproximadas, que podem ser satisfatórias dependendo do contexto do problema.