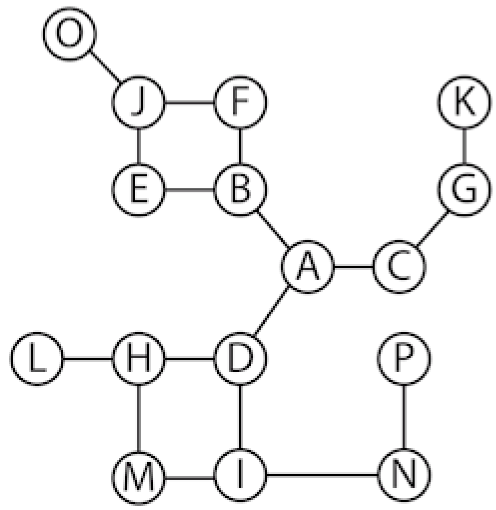
Estrutura de dados e algoritimos avançados

Nome: Ian costa dos Santos

Turma: 24E1_2

1. **Imagine a realização de uma busca em profundidade no grafo abaixo a partir do vértice “A”. Considere que a visitação entre vizinhos de um vértice será feita em ordem alfabética. Indique a ordem de visitação de todos os vértices no grafo.**



2. **Agora imagine a realização de uma busca em largura no mesmo grafo da questão anterior, novamente a partir do vértice “A”. Considere que a visitação entre vizinhos de um vértice será feita em ordem alfabética. Indique a ordem de visitação de todos os vértices no grafo.**

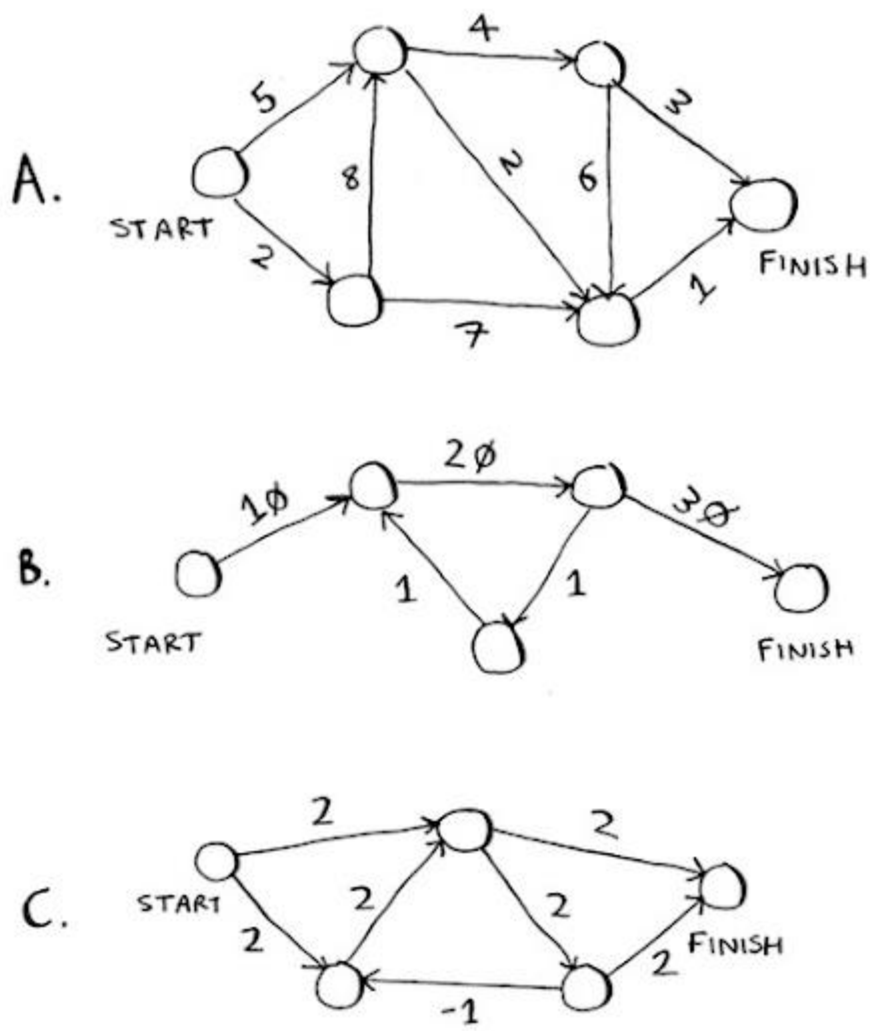
   Visitamos o primeiro vizinho de A em ordem alfabética, que é B.
   Do vértice B, seguimos para o vizinho E (primeiro em ordem alfabética).
   De E, como não há outros vizinhos que ainda não foram visitados, voltamos a B.

Do vértice B, o próximo vizinho em ordem alfabética que ainda não foi visitado é F.
De F, voltamos a B porque não há outros vizinhos que ainda não foram visitados.
Do vértice B, seguimos para J (próximo vizinho em ordem alfabética).
De J, visitamos O (primeiro em ordem alfabética).
De O, voltamos a J e, em seguida, a B, pois não há mais vizinhos não visitados.
Voltamos ao vértice A e seguimos para o próximo vizinho, que é C.
De C, visitamos D (primeiro em ordem alfabética).
De D, seguimos para H (primeiro em ordem alfabética).
De H, visitamos L (primeiro em ordem alfabética).
De L, voltamos a H e depois a D.
De D, o próximo vizinho é I (ordem alfabética).
De I, visitamos M (primeiro em ordem alfabética).
De M, voltamos a I e depois a D.
De D, o próximo vizinho é N.
De N, voltamos a I e depois a D.
Voltamos ao vértice C e seguimos para o próximo vizinho, que é G.
De G, voltamos a C e depois a A.
Do vértice A, o último vizinho é P.

A ordem de visitação dos vértices é, então:

A→B→E→F→J→O→C→D→H→L→I→M→N→G→P

5. Em cada um dos grafos abaixo, qual é o peso total do caminho mais curto do vértice “START” ao vértice “FINISH”?



<u>Grafo A:</u>

O caminho mais curto parece ser: START -> (peso da aresta 2) -> Nó -> (peso da aresta 7) -> Nó -> (peso da aresta 1) -> FINISH. Somando esses pesos, temos um peso total mínimo do caminho de START para FINISH como 2+7+1=10

<u>Grafo B:</u>

Existe apenas um caminho de START para FINISH que passa por três arestas com pesos em sequência: 10+20+30=60

<u>Grafo C:</u>

O caminho mais curto parece ser: START -> (peso da aresta 2) -> Nó -> (peso da aresta -1) -> Nó -> (peso da aresta 2) -> Nó -> (peso da aresta 2) -> FINISH. Somando esses pesos, temos um peso total mínimo do caminho de START para FINISH como 2+(−1)+2+2=5

7. **Use a notação Big O para descrever a complexidade da função implementada na questão 4, justificando sua resposta.**

Usando a notação Big O pode ser descrita como O(V + E), onde V é o número de vértices no grafo e E é o número de arestas.

No pior caso, a função visita todos os vértices e todas as arestas do grafo até encontrar o destino desejado ou até esgotar todas as possibilidades de caminho. Isso ocorre quando o grafo é um grafo completo, ou seja, cada vértice está conectado a todos os outros vértices. Nesse caso, a função precisa percorrer todos os vértices e todas as arestas, resultando em uma complexidade de O(V + E).

8. **Use a notação Big O para descrever a complexidade da função implementada na questão 6, justificando sua resposta.**

Usando a notação Big O como O(V + E), onde V é o número de vértices no grafo e E é o número de arestas. Aqui está a justificativa:

Visita de Vértices: A função utiliza uma busca em largura (BFS) para encontrar o caminho mais curto entre dois vértices em um grafo não ponderado. Durante a BFS, cada vértice é visitado no máximo uma vez. Portanto, o número de operações para visitar todos os vértices é proporcional ao número de vértices no grafo, ou seja, O(V).

Processamento de Arestas: Para cada vértice visitado, a função processa seus vizinhos (arestas) para determinar o caminho mais curto. Em um grafo não ponderado, o processamento das arestas não envolve cálculos complexos de pesos, então o número total de operações para processar os vizinhos de todos os vértices é proporcional ao número de arestas, ou seja, O(E).

Portanto, somando as operações para visitar todos os vértices (O(V)) com as operações para processar os vizinhos (O(E)), a complexidade total da função caminho_mais_curto em um grafo não ponderado é O(V + E)