

Aluno: Igor Xisto Ferreira

1) Determine os dois alunos que, provavelmente, possuem o melhor desempenho e defina o tipo de relação acadêmica existente entre eles na turma.

Com base na análise do grafo invertido e aplicação do BFS (busca em largura), os dois alunos mais alcançados foram o 10 e o 19. Isso indica que muitos outros alunos dependem deles academicamente, seja para ajuda com dúvidas, estudos ou trabalhos. A relação entre eles é direta, pois existe uma aresta $10 \rightarrow 19$, e indireta, pois o aluno 19 também influencia outros que chegam até ele via o aluno 10.

2) Explique, em suas próprias palavras, se neste caso podemos considerar tanto as arestas de entrada direta quanto os caminhos indiretos na definição de alcance.

Sim. A análise de influência acadêmica deve considerar não apenas relações diretas (arestas de entrada), mas também caminhos indiretos. Por exemplo, se o aluno 3 ajuda o 2 e o 2 ajuda o 1, então o 3 influencia indiretamente o 1. No código, foi usado o grafo invertido + BFS porque, no grafo original, as setas apontam de quem depende para quem é influente. Ao inverter essas arestas, é possível percorrer, a partir de um aluno, todos os que ele influencia, direta ou indiretamente. O fecho transitivo, por outro lado, mostra de quem um aluno depende, o que seria útil para saber sua "base de conhecimento", mas não sua influência.

3) O que é grafo invertido? Descreva como construí-lo a partir da lista de arestas.

O grafo invertido é aquele em que todas as arestas têm sua direção trocada. Se no grafo original há uma aresta $(A \rightarrow B)$, no grafo invertido haverá uma aresta $(B \rightarrow A)$. Isso é útil para saber "quem chega até alguém", ou seja, quem influencia um determinado aluno. No código, a função `grafoInvertido` constrói essa estrutura usando um `Map<Integer, List<Integer>>`.

4) Quais mudanças conceituais ocorrem na interpretação dos vizinhos ao usar o grafo invertido?

No grafo original, uma aresta $A \rightarrow B$ significa que A depende de B (ou seja, B influencia A). Assim, os vizinhos de A representam os alunos dos quais A depende diretamente.

Ao inverter o grafo, as arestas se tornam $A \leftarrow B$. Agora, os vizinhos de A representam os alunos que são influenciados por A. Portanto, no grafo invertido, os vizinhos de um vértice são os alunos que ele influencia diretamente.

Essa inversão é importante quando queremos analisar quem exerce mais influência (quem tem mais vizinhos no grafo invertido) ou simular a propagação de conhecimento, por exemplo usando BFS. Portanto, a vizinhança deixa de representar quem um aluno depende e passa a representar quem ele influencia.

5) Implemente o BFS que, dado um vértice v no grafo invertido, conte quantos nós são visitados.

A função `bfs(Map<Integer, List<Integer>> grafo, int alvo)` implementa essa lógica: ela usa uma fila (Queue) para explorar vértices e um Set para marcar quais já foram visitados. O algoritmo começa pelo vértice alvo e percorre seus predecessores no grafo invertido. A função retorna a quantidade de vértices que conseguem alcançá-lo (exceto ele mesmo).

6) Defina o fecho transitivo de um grafo e apresente o algoritmo de Warshall (versão booleana).

O fecho transitivo mostra todos os pares de alunos onde existe um caminho (direto ou indireto) de dependência acadêmica. No caso, se o aluno A depende de B, e B depende de C, então A depende indiretamente de C. No código, foi usada a versão booleana do algoritmo de Warshall, com uma matriz de adjacência `matriz[i][j] = 1` indicando que o aluno depende (direta ou indiretamente) do aluno j. Foram utilizados booleanos porque o que importa é a existência do caminho, não seu custo ou quantidade de passos. Isso é útil para saber a quantidade e abrangência do conhecimento que um aluno consome — diferente do grafo invertido + BFS, que mede a influência exercida.

7) Suponha que, após executar as buscas, os alunos A e B tenham contagens máximas de “quem os alcança”. Como você justifica que esses sejam os melhores desempenhos acadêmicos?

No grafo original, esses alunos são fontes de conhecimento, ou seja, outros dependem deles. Isso sugere que A e B possuem alto desempenho, já que são frequentemente procurados como referência acadêmica.

8) Se A e B estiverem numa mesma componente fortemente conectada, o que isso diz sobre a relação de dependência entre eles?

Isso significa que existe um caminho de A para B e de B para A. Em termos práticos, eles não apenas ajudam outros, mas também aprendem entre si. Essa situação reflete uma relação de dependência mútua, onde o aprendizado é colaborativo e recíproco.

9) Além de contar predecessores, que outras métricas de centralidade poderiam indicar “influência” num grafo de dependências?

Nesta questão, foram analisadas diferentes formas de medir a influência de um aluno na rede de dependência acadêmica. As métricas foram aplicadas sobre o grafo invertido, onde uma aresta indica que um aluno influencia outro (ou seja, o vértice de origem da aresta é a fonte de conhecimento).

- **Centralidade de Grau (grau de saída no grafo invertido):** Mede quantos alunos um determinado vértice influencia diretamente. No grafo invertido, corresponde à quantidade de arestas que saem do aluno, ou seja, quantos ele alcança diretamente. Alunos com maior grau de saída no grafo invertido são os que mais outros alunos

dependem diretamente deles.

Exemplo da saída do código:

Aluno 20 e Aluno 29: grau de entrada = 6

Aluno 8 e Aluno 25: grau de entrada = 5

- **Centralidade de Aproximação:** Calculada como a soma das distâncias mínimas até os demais alunos no grafo invertido. Mede quão rapidamente um aluno pode influenciar toda a rede, considerando caminhos indiretos. Quanto menor a soma, maior a capacidade de disseminar conhecimento.

Exemplo da saída do código:

Aluno 29: soma das distâncias = 45

Aluno 8: soma das distâncias = 50

- **Centralidade de Intermediação:** Quantifica quantas vezes um aluno aparece nos caminhos mínimos entre pares de alunos. Representa a importância como ponte entre diferentes partes da rede. Pode indicar alunos essenciais para a conexão entre grupos de dependência acadêmica.

Não foi implementada.

- **Centralidade de Autovetor:** Associa mais valor a alunos que influenciam outros alunos também influentes. Mede a importância qualitativa das conexões. Um aluno com alta centralidade de autovetor é aquele que está bem posicionado em relação aos demais influentes.

Não foi implementada.

10) Em qual situação você preferiria usar fecho transitivo em vez de inverter + BFS? E vice-versa? (pensando em grafos esparsos vs. densos, em limites de memória e em facilidade de implementação.)

Usar fecho transitivo (Warshall): O fecho transitivo é preferível quando o grafo é pequeno e denso, pois o algoritmo tem custo $O(n^3)$. Também é ideal quando se precisa responder a muitas perguntas sobre alcance entre todos os pares de vértices rapidamente.

Usar grafo invertido + BFS: Esta abordagem é ideal para grafos esparsos e grandes. É mais eficiente quando se quer apenas responder sobre um vértice específico. No código, foi usado o algoritmo de Warshall para ver todos os alcances e o BFS quando o foco é saber quem alcança um único aluno.

11) Qual é a limitação de usar somente o grau de entrada direta (Grau de Entrada) para medir 'influência'?

O grau de entrada conta apenas relações diretas, ignorando toda a influência indireta que um aluno pode ter. Por exemplo, um aluno pode ser altamente influente por meio de intermediação (ajuda um que ajuda vários), mas isso não aparece no grau de entrada simples.

12) Como a existência de ciclos no grafo pode distorcer essa medida?

Ciclos geram uma contagem inflada, pois os mesmos nós podem ser revisitados em diferentes caminhos. Se não tratados corretamente (por exemplo, não marcando o nó como visitado no BFS), isso pode causar *overcounting* ou *loops* infinitos. No código, um Set foi usado para evitar visitar os mesmos nós novamente.

**13) Modifique o grafo para adicionar pesos às arestas (nível de dependência).
Como você ajustaria o cálculo de alcance indireto?**

Ao adicionar pesos (ex: 1 2 4.5), é interpretado que certas dependências são mais fortes. Agora o peso representa o nível de conhecimento de certo aluno que se pede ajuda. Para medir o "alcance", não basta saber se é possível chegar, agora se quer o menor custo de chegada. Foi usado o algoritmo de Dijkstra no código para isso, armazenando em um `Map<Integer, List<int[]>>` as arestas ponderadas e calculando os menores caminhos a partir de um vértice somando os pesos das arestas.