Aluno(a): ____________________

Primeira Avaliação (Valor: 10,0)

1. [Valor: 2,0] Programas podem ser executados mais rapidamente se certas instruções forem processadas concorrentemente. É importante não executar uma instrução que requeira resultados de outras instruções ainda não executadas. A dependência de instruções em relação a outras instruções pode ser representada por um grafo direcionado. Cada instrução é representada por um vértice e existe uma aresta de um vértice u a um vértice v se a instrução u deve terminar sua execução antes da instrução v começar sua execução. Este grafo é chamado de *grafo de precedência.*

   (a) [Valor: 1,0] Considere o seguinte trecho de código sequencial (cada linha representa uma instrução s_i). Desenhe o grafo de precedência. Represente-o por matriz de adjacência e por lista de adjacência.

   $$\begin{aligned} s_1&: \quad a = 0; \\ s_2&: \quad b = 1; \\ s_3&: \quad c = a + 1; \\ s_4&: \quad d = a + b; \\ s_5&: \quad e = d + 1; \\ s_6&: \quad f = c + d; \end{aligned}$$

   (b) [Valor: 0,5] Uma ordem para executar estas instruções seria: $s_1, s_2, s_3, s_4, s_5, s_6$. Apresente mais três ordenações topológicas para este problema.

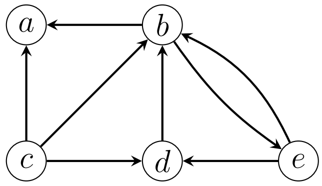
   (c) [Valor: 0,5] Todo grafo direcionado possui ao menos uma ordenação topológica? Explique.

2. [Valor: 2,0] O comportamento de algumas pessoas pode influenciar o pensamento de outras. O grafo de influência considera que cada pessoa é um vértice e existe uma aresta de u para v se o indivíduo v é influenciado pelo indivíduo u.

   (a) [Valor: 1,0] Seja *coitado* o indivíduo incapaz de influenciar alguém e *manipulador* o indivíduo que não é influenciado por qualquer outro indivíduo. Dado um grafo $G = (V, E)$ representado por lista de adjacências, informe como podemos determinar todos os coitados e todos os manipuladores de G.

   (b) [Valor: 0,5] Seja *partido* um conjunto maximal de indivíduos, tal que para todo u e v deste conjunto, u influencia (direta ou indiretamente) v e vice-versa. Informe que problema estudado em grafos poderia resolver o problema de determinar todos os partidos.

   (c) [Valor: 0,5] Dado o grafo a seguir, indique quais são os coitados, manipuladores e os partidos.

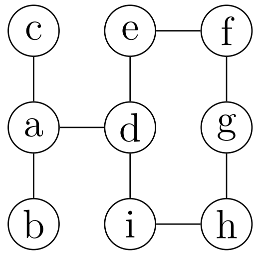


3. [Valor: 2,0] Considere o grafo $G = (V, E)$ a seguir e responda.

   (a) [Valor: 1,0] Ao usarmos o algoritmo para identificar pontos de articulação e começando pelo vértice a, descobrimos que os vértices a e d são pontos de articulação. Eles são pontos de articulação pelo mesmo motivo? Explique como o algoritmo consegue identificar tais pontos (explique as propriedades que fazem com que um vértice v seja considerado ponto de articulação).

(b) [Valor: 0,5] Para cada vértice $v \in V$, determine o valor de $low[v]$ (tempo de descoberta do acestral mais antigo de v que pode ser alcançado em um passo a partir de um descendente de v). Comece pelo vértice h.

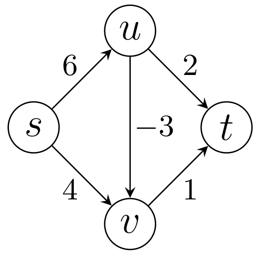
(c) [Valor: 0,5] Sabe-se que existe um caminho $a \rightsquigarrow c$ e um caminho $c \rightsquigarrow a$. O mesmo pode ser dito sobre os pares (d, e), (e, f), (f, g), (g, h), (h, i) e (i, d). Por que estas arestas não são pontes? Qual a diferença de (a, c) para estas arestas?



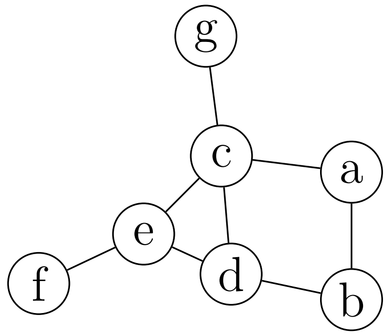
4. [Valor: 2,0] Responda as seguintes questões sobre caminhos mínimos. Considere o vértice s como vértice de início.

   (a) [Valor: 1,0] O que aconteceria se usássemos o algoritmo de Dijkstra no grafo a seguir? Explique por que arestas com custos negativos (mesmo que não pertençam a um ciclo de custo negativo) podem fazer com que o algoritmo de Dijkstra falhe.

   (b) [Valor: 1,0] E se usássemos o algoritmo de Bellman-Ford? É verdade que, independentemente da ordem em que relaxamos as arestas, após a primeira iterção teríamos o caminho mínimo de s a v, na segunda teríamos o caminho de s a u, e por fim, na terceira teríamos o caminho mínimo de s a t? Explique.



5. [Valor: 2,0] A distância $\delta(s, v)$ entre os vértices s e v em um grafo não ponderado e não orientado é o número de arestas em um caminho mínimo conectando eles, ou seja, o comprimento do menor caminho entre s e v. A *excentricidade* de um vértice v é a maior distância entre v e qualquer outro vértice. O *centro* de um grafo é o conjunto de todos os vértices com excentricidade mínima. Por exemplo, no grafo a seguir, os vértices a, b, f e g possuem excentricidade três; os vértices c, d e e possuem excentricidade dois. Portanto, o centro do grafo é o conjunto de vértices $\{c, d, e\}$. Descreva um algoritmo que, dado um grafo não ponderado e não orientado, devolve como resposta o centro do grafo.



Boa Prova!!!