

MC558 Projeto e Análise de Algoritmos
Lista de exercícios 2 — data de entrega: 24 de maio

Instruções:

- (i) As respostas devem ser digitadas usando qualquer editor/formatador (sugiro \LaTeX , se você souber usar). As submissões devem ser feitas em formato pdf no Google Classroom; você pode anexar figuras, mas gere um único arquivo. Soluções que não respeitem estas condições receberão **nota ZERO**.
- (ii) Todas as respostas devem ter justificativas (corretude e/ou complexidade), a menos que a questão diga explicitamente que não são necessárias.
- (iii) Você pode usar qualquer resultado ou algoritmo visto em aula. Conforme o caso, enuncie o resultado ou escreva qual é a complexidade do algoritmo, caso seja necessário na análise de complexidade.
- (iv) Em qualquer questão que exija um pseudo-código complicado, explique sua ideia antes de escrevê-lo (no máximo uma página, mas isto provavelmente é muito dependendo da questão). Outra forma é você explicar em alto nível o que faz cada trecho de código.

Sua explicação deve ser boa o suficiente para me convencer que o algoritmo funciona (inclua provas de resultados auxiliares, se necessário). Note que explicar bem não é o mesmo que explicar muito. Soluções que tenham pseudo-códigos complicados, mas sem nenhuma tentativa razoável de explicação não serão consideradas.
- (v) Os pseudo-códigos devem ter estilo semelhante aos apresentados em aula ou que estão no livro do CLRS. Pseudo-códigos com trechos de linguagem de programação como C ou Python **não** serão aceitos. Você pode implementar um programa para resolver a questão, se quiser, mas **não** aceitarei como resposta um copy-and-paste do código sob nenhuma hipótese.
- (vi) Em um pseudo-código você pode devolver diretamente um conjunto (e.g., escreva “devolva Q ”). Você pode usar instruções em português também, e.g., “devolva os vértices da árvore T ” ou “ordene a sequência X ” ou “execute DFS sobre o grafo G ”. Há muitas situações em que é razoável usar uma instrução deste tipo. Tenha em mente que uma instrução deste tipo consome uma certa quantidade de tempo que você deve analisar e deve ser razoavelmente óbvio (para mim) que ela pode ser executada no tempo descrito.
- (vii) Em qualquer questão que exija uma descrição de um algoritmo em alto nível (sem pseudo-código), descreva-o de maneira clara e precisa em português. A descrição dos seus passos deve ter detalhes suficientes para eu poder concluir que o algoritmo tem a complexidade exigida.
- (viii) Se você usar alguma notação que não está nos slides ou no CLRS, você deve explicar precisamente o que representa. Não tenho como saber toda notação usada em outras fontes. É esperado também que ninguém invente uma notação para algo que já tem uma notação definida e que foi bastante usada nas aulas.

- (25 pontos) Modifique o pseudo-código do algoritmo de busca em profundidade **apresentado em aula ou do CLRS** (supondo que o grafo de entrada G é orientado) para imprimir cada aresta (u, v) juntamente com seu tipo (aresta da árvore, de avanço, de retorno ou de cruzamento). A complexidade do DFS modificado ainda deve ser $O(V + E)$. **Observação:** basta escrever o pseudo-código, sem explicação ou prova de corretude; se você usou variáveis que não estavam no pseudo-código original de DFS, explique o que representam. **Você deve escrever o pseudo-código inteiro.**
- (75 pontos) Seja G um **grafo orientado acíclico**. Suponha que cada aresta $(u, v) \in E[G]$ tem uma cor $\text{cor}(u, v)$ que pode ser **azul** ou **vermelha**. Um caminho P em G é **válido** se **não possui arestas consecutivas de cor vermelha**. Veja um exemplo abaixo:

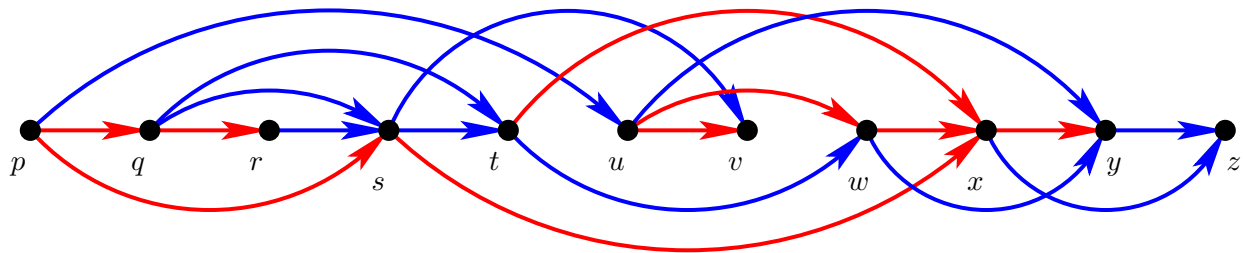


Figura 1: Para quem tem dificuldades de distinguir as cores, as **arestas vermelhas** são: (p, q) , (q, r) , (p, s) , (s, x) , (t, x) , (u, v) , (u, w) , (w, x) e (x, y) . Caminhos de comprimento zero ou um são sempre válidos. Os caminhos (q, s, t, x, z) e (p, s, t, w, y) também são válidos. Já o caminho (p, q, s, t, x, y, z) não é válido pois (t, x) e (x, y) são arestas consecutivas de cor vermelha neste caminho.

Nesta questão, você deve projetar um **algoritmo linear** que para cada vértice $u \in V[G]$, devolve o **número de caminhos válidos que começam em u** .

Definição. Defina $\text{azul}[u]$ (respectivamente, $\text{verm}[u]$) como o número de caminhos válidos com início em u cuja primeira aresta tem cor azul (respectivamente, vermelha). Note que o caminho trivial válido (u) não contribui para nenhum desses valores.

- (a) (valor = 5 pontos) Para cada vértice i do grafo acima, indique os valores $\text{azul}[i]$ e $\text{verm}[i]$ (alguns valores estão preenchidos).

i	p	q	r	s	t	u	v	w	x	y	z
$\text{azul}[i]$						2	0	2	1	1	0
$\text{verm}[i]$						4	0	2	2	0	0

(b) (valor = 30 pontos) Descreva uma recorrência que relaciona $\text{azul}[u]$ em função de $\text{azul}[v]$ e $\text{verm}[v]$ para $v \in \text{Adj}[u]$. **Dica:** Imagine que conhecemos os valores $\text{azul}[v]$ e $\text{verm}[v]$ para cada $v \in \text{Adj}[u]$. Como podemos calcular $\text{azul}[u]$? Além disso, note também que uma aresta (u, v) de cor azul contribui com uma unidade para o valor $\text{azul}[u]$ pois é um caminho válido de comprimento um.

Descreva agora uma recorrência que relaciona $\text{verm}[u]$ em função de $\text{azul}[v]$ e $\text{verm}[v]$ para $v \in \text{Adj}[u]$. **Dica:** pense da mesma forma que antes, mas agora note que um caminho válido cuja primeira aresta (u, v) tem cor vermelha **não** pode usar uma aresta (v, w) de cor vermelha. Como acima, note que uma aresta (u, v) de cor vermelha contribui com uma unidade para o valor $\text{verm}[u]$ pois é um caminho válido de comprimento um.

(c) (valor = 30 pontos) Escreva um pseudo-código de um algoritmo de complexidade $O(V + E)$ que recebe um grafo orientado acíclico G representado por listas de adjacências e um vetor **cor** de cores e devolve um vetor $\text{val}[\]$ indexado por V tal que $\text{val}[u]$ é o número de caminhos válidos que começam em u para cada $u \in V[G]$. No exemplo da figura, $\text{val}[u] = 7$, $\text{val}[v] = 1$, $\text{val}[w] = 5$, $\text{val}[x] = 4$, $\text{val}[y] = 2$ e $\text{val}[z] = 1$.

(d) (valor = 10 pontos) Justifique a complexidade do seu algoritmo do item (c).

Observações: você pode supor que é possível determinar a cor de uma aresta em tempo $O(1)$; em particular, você pode fazer testes do tipo “se $\text{cor}(u, v) = \text{azul}$ ”.