# PROVA 1 DE MAC0328 ALGORITMOS EM GRAFOS

Observações gerais:

- Em todos os exercícios, *justifique a corretude e o consumo de tempo do seu algoritmo*. Não é preciso apresentar provas completas; bastam argumentos convincentes. Faz parte da sua tarefa avaliar o que constitui um "argumento convincente", bem como o que pode ser omitido por ser "óbvio".
- Nos exercícios em que se menciona *pseudocódigo*, requer-se nível de detalhe semelhante ao visto em aula. Nos demais, você pode usar como caixa preta os algoritmos já vistos, contanto que descreva o que devolvem e quanto tempo consomem.

[20 pontos] **Exercício 1.** Escreva um *pseudocódigo* para busca em profundidade em um **digrafo** $D = (V, A)$ que classifica todos os arcos de $D$ nos seguintes tipos: arcos de arborescência, arcos de retorno, arcos descendentes e arcos de cruzamento. Seu algoritmo deve consumir tempo $O(n + m)$. Neste exercício, não é preciso justificar a corretude.

[5 pontos] **Bônus 1.** Ganhe 5 pontos a mais se seu algoritmo diferenciar entre os dois tipos de arcos de cruzamento.

**Exercício 2.** Escreva um *pseudocódigo* para um algoritmo que, dado um **grafo** conexo $G = (V, E)$, encontra uma ponte $e \in E$ tal que um dos componentes de $G - e$ não possui pontes, ou determina que uma tal ponte não existe. Seu algoritmo deve consumir tempo $O(n + m)$.

[20 pontos] **Exercício 3.** Seja $G = (V, E)$ um **grafo**. Um subconjunto $T \subseteq V$ contendo exatamente 3 vértices é chamado um *triângulo* de $G$ se $\{u, v\} \in E$ para todos $u, v \in T$ distintos. Projete um algoritmo que computa o número de triângulos de um dado grafo $G$. Seu algoritmo deve consumir tempo $O(n^{\omega})$, onde $\omega := 2.38$. Você pode supor que tem acesso a um algoritmo que calcula o produto de duas matrizes $n \times n$ em tempo $O(n^{\omega})$.

[20 pontos] **Exercício 4.** Um **digrafo** $D = (V, A)$ é chamado *semiconexo* se, para quaisquer vértices distintos $u, v \in V$, vale que $u \rightsquigarrow v$ ou[1] $v \rightsquigarrow u$. Desenvolva um algoritmo $O(n + m)$ que determina se um dado digrafo é semiconexo.

[20 pontos] **Exercício 5.** Considere um digrafo $D = (V, A)$ em que cada arco $uv \in A$ pode falhar com probabilidade $p_{uv} \in (0, 1)$. Suponha ainda que os arcos falham ou funcionam independentemente.[2] Desenvolva um algoritmo eficiente que, dado tal digrafo $D$, probabilidades $p\colon A \to \mathbb{R}$ e vértices $r, s \in V$, encontra um caminho de $r$ a $s$ em $D$ que tem a menor probabilidade de falha.

---

*Data*: 7 de abril de 2015.

[1] Este "ou" é inclusivo.

[2] Formalmente, se $F_{uv}$ denota o evento em que o arco $uv \in A$ falha, suponha que a coleção de eventos $\{F_{uv} : uv \in A\}$ é mutuamente independente.