Universidade Federal de Sergipe – CCET – Departamento de Computação
Grafos e Algoritmos Computacionais - Turma T02 - 2024.1 – Segunda Prova
Professor: Breno Piva Ribeiro **Aluno:**

1) (valor: 2,0) Se um grafo direcionado G contém um caminho de u a v, e se PE(u) < PE(v) numa busca em profundidade de G, então v é um descendente de u na **floresta** de profundidade produzida. Prove ou dê um contra-exemplo. Obs.: PE(v) é a profundidade de entrada do vértice v.

2) (valor: 2,5) Existem dois tipos de wrestlers profissionais: “babyfaces” (“bonzinhos”) e “heels” (“malvados”). Para cada par de wrestlers profissionais pode ou não haver rivalidade entre eles. Suponha que nós temos n wrestlers profissionais e nós temos uma lista de r pares de wrestlers para os quais existe rivalidade. Explique como seria um algoritmo de complexidade de tempo $O(n + r)$ para determinar se é possível designar alguns wrestlers como “babyfaces” e os demais como “heels” de modo que toda rivalidade seja entre um “babyface” e um “heel”. Qual a propriedade que garante que seu algoritmo funciona? Prove esta propriedade.

3) (valor: 2,0) Seja (u, v) **uma** aresta de peso mínimo em um grafo conexo G. Mostre que (u, v) pertence a **alguma** árvore geradora mínima de G. Dica: pense em uma prova por contradição.

4) (valor: 2,0) Dê exemplo de um grafo G, para o qual $\chi(\alpha_{v,w}(G)) < \chi(\beta_{v,w}(G))$, sendo v,w um par arbitrário de vértices não adjacentes em G.

5) (valor: 2,5) Explique porque o critério de Edmonds-Karp para escolha do caminho aumentante em uma rede residual garante que o algoritmo de Ford-Fulkerson (listado abaixo) termine em tempo polinomial.

```
Algoritmo Ford–Fulkerson
Entrada: rede D(V, E) com capacidades c(e) para cada aresta e em E,
fonte s em V e sumidouro t em V
Saída: função f de fluxo em cada aresta
     F := 0
     para e em E faça
          f(e) := 0
     construir a rede residual D′(f)
     enquanto existir caminho v_1,…,v_k de s = v_1 a t = v_k em D′ faça
          F′ := min{c′(v_j, v_j+1) | 1 <= j <= k}
          para j = 1, …, k–1 faça
               se(v_j, v_j+1) é aresta direta então
                    f(v_j, v_j+1) := f(v_j, v_j+1) + F′
               senão
                    f(v_j+1, v_j) := f(v_j+1, v_j) – F′
          F := F + F′
          construir a rede residual D′(f)
```

Fim Algoritmo