

Observação: Não é necessário responder todas perguntas para alcançar 10 pontos.

Prova

Questão 1 (Fluxo máximo, 2.5pt)

Um colega defende a ideia de ignorar os arcos para atrás (backward edges) no algoritmo de Edmonds-Karp para encontrar o fluxo máximo. Ele afirma que fica mais simples, mais rápido, e a perda não é grande: o algoritmo resultante é uma 2-aproximação ao problema (i.e. nunca retorna um fluxo com valor menos que a metade do fluxo máximo). Ele tem razão? (Prova ou contra-exemplo.)

Questão 2 (Aproximação, 2.5pt)

Um segundo colega melhorou o algoritmo de Cristofides como segue. Ele calcula um emparelhamento perfeito de peso mínimo entre todos vértices. (Para ficar mais simples vamos supor que o número de vértices no grafo é par.) Depois ele exclui as arestas selecionados do grafo e calcula um outro emparelhamento perfeito de peso mínimo entre todos vértices. Na união dos dois emparelhamentos todo vértice possui grau 2, logo o grafo é Euleriano e o resto do algoritmo funciona igual. Além disso, pelo mesmo argumento do algoritmo de Cristofides, cada emparelhamento pesa no máximo $OPT/2$, logo a rota final é ótima. Ele tem razão? (Prova ou contra-exemplo.)

Questão 3 (Algoritmo de Dijkstra, 2.5pt)

Para encontrar o caminho mais curto entre vértices s e t , o algoritmo de Dijkstra pode parar ao processar o vértice t (i.e. no momento em que o vértice t é retirado da fila de prioridade). Considera os seguintes dois critérios de parada mais agressivos:

- Termina caso um vértice v com $d_v = d_t$ é processado. (Aqui d_v é a distância atual do vértice $v \in V$.)
- Termina caso o vértice t é atualizado (i.e. durante o processamento de algum vértice $v \neq t$ uma operação `decreasekey` no vértice t é chamada).

Eles são corretos? (Prova ou contra-exemplo.)

Questão 4 (Caminhos disjuntos, 2.5pt)

Dado um grafo direcionado queremos achar o número máximo de caminhos disjuntos de arcos entre dois vértices s e t . Propõe um algoritmo eficiente para resolver este problema. (Argumento de corretude & análise de complexidade. A pontuação depende da complexidade.)

Questão 5 (Emparelhamentos, 2.5pt)

Uma colega propõe um algoritmo randomizado para resolver o problema de emparelhamentos de cardinalidade máxima em grafos bipartidos. Enquanto existe pelo menos um par de vértices livres com uma aresta entre eles, o algoritmo seleciona uma aresta candidata que satisfaz essa condição aleatoriamente. Ela afirma que isso é uma 2-aproximação para o problema (i.e. nunca retorna menos que a metade do emparelhamento máximo). Ela tem razão? (Prova ou contra-exemplo.)

Questão 6 (Heaps, 2.5pt)

Compara um heap binário com heaps ociosos (realização e complexidade das principais operações).

Sucesso!