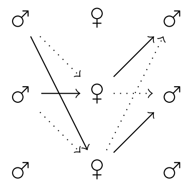
Universidade Federal do Rio Grande do Sul
Instituto de Informática
Departamento de Informática Teórica

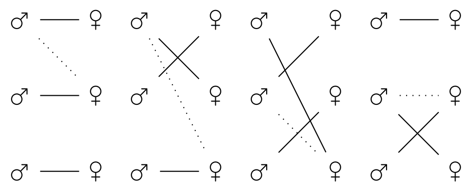
CMP601 – Teoria da Computação e Algoritmos
2015/1
Profa. Leila Ribeiro, Prof. Marcus Ritt

Lista de soluções 1

Questão 1
As preferências são dados por



(primeira camada: preferência de homens, segunda camada: preferência de mulheres).
Observe que o terceiro homem e a primeira mulher não tem preferências definidas. Eles não importam no contra-exemplo que segue. Com as preferências acima a seguinte sequência de trocas que leva a um ciclo é possível



Questão 2
O algoritmo funciona da mesma maneira que no caso *balanceado* (com n homens e mulheres). Em particular, como os homens propõem, uma mulher que está com um noivo nunca mais volta a ser solteira. Logo, o algoritmo termina sem homem solteiro, porque depois de ter proposto para $n+1$ mulheres, todas possuem um noivo, uma contradição. Logo, diferente do caso balanceado, sabemos que no máximo n mulheres recebem uma proposta.
Por um argumento idêntico com o caso balanceado uma instabilidade do tipo 1 não pode ocorrer. Agora considera a mulher solteira w que não recebeu nenhuma proposta. Caso existe uma instabilidade do tipo 2, um homem h noivo de uma mulher w' prefere w. Mas então ele já fez uma proposta para w, uma contradição com o fato que w não recebeu nenhuma proposta.
(O argumento pode ser facilmente generalizado para qualquer número de mulheres $> n$.)

Questão 3
Uma abordagem natural é tentar usar uma busca em largura (ou profundidade) no grafo. Toda aresta (u, v) que não faz parte da árvore de busca fecha um ciclo. Para determinar o comprimento do ciclo, temos que encontrar o (único) caminho entre u e v na árvore de busca, dado pelo menor ancestro em comum. No final, podemos retornar o menor ciclo encontrado.
Porém essa solução não retorna todos ciclos, e em particular não precisa encontrar o menor ciclo. Um exemplo é mostrado na Figura 1. A busca a partir de v encontra os ciclos $ywvx$ e $ywvxz$, mas não o ciclo xyz.
Podemos observar que a busca em largura em geral encontra pelo menos um ciclo mais curto que contém o vértice de origem v. (Porém ela não encontra todos ciclos, nem todos ciclos mais curtos, e.g. num grafo completo.)

Proposição 1
A busca em largura a partir de um vértice v encontra um dos ciclos mais curtos que contém v.

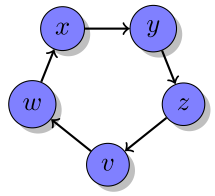
Prova. Considere um ciclo mais curto que contém v. A distância mínima entre v e um vértice u do ciclo e igual a distância mínima entre v e u no ciclo (*). Caso contrário podemos obter um ciclo mais curto, uma contradição.
A busca em largura (BL) encontra um ciclo, caso uma aresta não-árvore conecta dois vértices já explorados. Supõe agora que a BL não encontra nenhum ciclo mais curto. Vamos considerar dois casos.

Primeiramente supõe que o ciclo mais curto tem comprimento $2k+1$. Neste caso tem dois vértices w, x de distância k de v. Por (*) existem caminhos de v para w e x de comprimento k na árvore. Como a aresta wx faz parte do ciclo, ao processar w ou x vamos encontrar um ciclo de comprimento $2k+1$. Agora supõe que o ciclo mais curto tem comprimento $2k$. Neste caso tem dois vértices w, x de distância $k-1$ e k de v. Por (*) existe um caminho vw de comprimento $k-1$ e outro caminho vx de comprimento k de v na árvore. Novamente, como a aresta wx faz parte da árvore, ao processar w ou x vamos encontrar um ciclo de comprimento $2k$.

■

Logo, uma solução é buscar em largura a partir da cada vértice $v \in V$. Isso custa tempo $O(n(n+m)) = O(nm)$. (Existem algoritmos mais eficientes.)

Questão 4
A proposição é falsa. Um contra-exemplo é



A componente estranha de x é $\{x, z, v\}$, a de z é $\{z, w, x\}$.

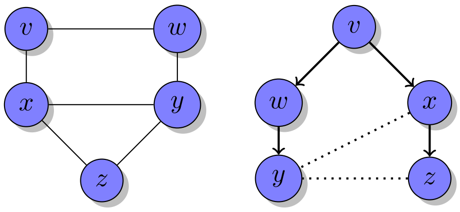


Figura 1: Exemplo de uma busca em largura que não encontra o menor ciclo. Esquerda: grafo e vértice inicial v. Direita: árvore de busca.