

Troca de e por e'

Leandro Teófilo (teofilo.ifto@gmail.com)

Nesta ata, será mostrada, a partir de códigos apresentados pelo aluno Rafael Werneck durante a aula da disciplina MO405/2012s1 (Teoria dos Grafos), a resolução das seguinte proposições:

Proposição 1: Sendo T, T' duas árvores geradoras de um grafo conectado G e $e \in E(T) - E(T')$, existe uma aresta $e' \in E(T') - E(T)$, tal que $T - e + e'$ é uma árvore geradora de G .

Proposição 2: Sendo T, T' duas árvores geradoras de um grafo conectado G e $e \in E(T) - E(T')$, existe uma aresta $e' \in E(T') - E(T)$, tal que $T' + e - e'$ é uma árvore geradora de G .

Código

Código feito em Python. Contém os seguintes arquivos:

- Trocas_e_elinha.zip
 - troca_e_elinha.py
 - troca_elinha_e.py
 - arvores

O arquivo **troca_e_elinha.py** prova a Proposição 1. O arquivo **troca_elinha_e.py** prova a a Proposição 2. O arquivo arvores é um arquivo de entrada, servindo tanto para o primeiro quanto para o segundo programa.

troca_e_elinha.py

Bem estruturado o código, define o que é um grafo, uma aresta e um vértice, seguindo modelo do livro texto para implementação.

Através do arquivo de entrada, o programa inicializa as duas arvores como dois grafos T1 e T2, representando assim, T e T' da proposição 1). Após isso, o programa solicita do usuário a aresta a ser retirada de T1, verificando a necessidade de a aresta não pertencer simultaneamente a T1 e T2. Tão logo seja informada uma aresta válida, o programa retira a aresta de T1.

O programa então, realiza uma busca na segunda árvore(T2) por uma aresta que adicionada a T1(no momento não necessariamente uma árvore), mantenha as características necessárias para que T1 seja uma árvore geradora, provando assim a Proposição 1.

troca_elinha_e.py

Bem estruturado o código, assim como o primeiro código, define o que é um grafo, uma aresta e um vértice, seguindo modelo do livro texto para implementação.

Através do arquivo de entrada, o programa inicializa as duas arvores como dois grafos T1 e T2, representando assim, T e T' da proposição 2). Após isso, o programa solicita do usuário a aresta de T1 a ser incluída em T2, verificando a necessidade de a aresta não pertencer simultaneamente a T1 e T2. Tão logo seja informada uma aresta válida, o programa insere a aresta em T2.

O programa então, realiza uma busca na segunda árvore(T2) por uma aresta que deverá ser excluída para que T2 mantenha as características necessárias de uma árvore geradora, provando assim a Proposição 2.