IST - LMAC, MEBiom

Elementos de Programação

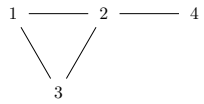
15 de Dezembro de 2017 Ficha 3A Duração: 15m

Número: _____ **Nome:** _____

Considere grafos (não-dirigidos), onde se assume que os nós são identificados pelos naturais $1, \ldots, k$ para um grafo com `k` nós, com as seguintes operações:

- `nodes(k)`: grafo com `k` nós e sem arestas;
- `edge(g,i,j)`: adiciona ao grafo `g` uma aresta entre os nós `i` e `j`;
- `edgeQ(g,i,j)`: `True` se existe no grafo `g` uma aresta entre os nós `i` e `j`, e `False` caso contrário;
- `delnode(g,i)`: grafo que resulta de `g` eliminando o nó `i` e todas as arestas que o envolvem;
- `dim(g)`: número de nós do grafo `g`.

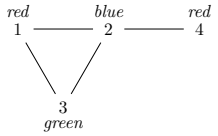
Pretende-se, em *Python*, representar cada grafo como uma lista de listas da forma $[w_1, w_2, \ldots, w_k]$ onde `k` é o número de nós do grafo e cada w_i é a lista dos nós que partilham uma aresta com o nó `i`. Nomeadamente, $[[2,3],[1,3,4],[1,2],[2]]$ deverá ser uma representação do grafo



a) Apresente implementações eficientes apenas para as operações `edgeQ` e `delnode`.

(vsff)

b) Uma coloração de um grafo é uma atribuição de cores aos nós do grafo de forma a que cada aresta ligue nós de cores distintas, como exemplificado.



Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `iscoloringQ` que recebendo um grafo `g`, e a lista de cores que desejamos atribuir aos nós do grafo, por ordem, devolve `True` se se tratar de uma coloração de `g`, e `False` caso contrário.

Nomeadamente, para o grafo acima colorido com a respectiva lista de cores [”`red`”, ”`blue`”, ”`green`”, ”`red`”] a resposta deverá ser `True`.

IST - LMAC, MEBiom

Elementos de Programação

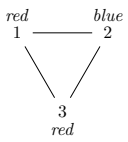
15 de Dezembro de 2017 Ficha 3B Duração: 15m

Número: _____ **Nome:** _____

Considere grafos (não-dirigidos) coloridos, onde se assume que os nós são identificados pelos naturais `1`, . . . , `k` para um grafo com `k` nós, com as operações:

- `nodes(w)`: dada uma lista de cores `w`, constrói o grafo colorido com `len(w)` nós, sem arestas, onde o nó `i` tem cor `w[i-1]`;
- `edge(g,i,j)`: adiciona ao grafo colorido `g` uma aresta entre os nós `i` e `j`;
- `edgeQ(g,i,j)`: `True` se existe no grafo colorido `g` uma aresta entre os nós `i` e `j`, e `False` caso contrário;
- `delnode(g,i)`: grafo colorido que resulta de `g` eliminando o nó `i` e todas as arestas que o envolvem;
- `colorofnode(g,i)`: cor associada ao nó `i` no grafo colorido `g`;
- `dim(g)`: número de nós do grafo `g`.

Pretende-se, em *Python*, representar cada grafo colorido como um par (`w`, `m`) onde `w` é a lista de cores associadas aos nós do grafo e `m` é uma matriz quadrada cujo número de linhas e colunas é o número de nós do grafo e em que cada entrada `m`[`i` − `1`][`j` − `1`] é `1` se existe uma aresta a ligar os nós `i` e `j`, e `0` caso contrário. E.g., ([”`red`”, ”`blue`”, ”`red`”], [[`0`, `1`, `1`], [`1`, `0`, `1`], [`1`, `1`, `0`]]) deverá ser uma representação do grafo



a) Apresente implementações eficientes apenas para as operações `edge` e `delnode`.

(vsff)

b) Um grafo colorido diz-se *legal* se cada aresta do grafo liga nós de cores distintas.

Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `legalQ` que recebendo um grafo colorido `g`, devolve `True` se se tratar de um grafo colorido legal, e `False` caso contrário. Nomeadamente, para o grafo colorido da página anterior a resposta deverá ser `False`.

IST - LMAC, MEBiom

Elementos de Programação

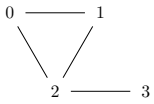
15 de Dezembro de 2017 Ficha 3C Duração: 15m

Número: _____ **Nome:** _____

Considere grafos (não-dirigidos), onde se assume que os nós são identificados pelos naturais $0, \ldots, k-1$ para um grafo com `k` nós, com as seguintes operações:

- `nodes(k)`: grafo com `k` nós e sem arestas;
- `edge(g,i,j)`: adiciona ao grafo `g` uma aresta entre os nós `i` e `j`;
- `edgeQ(g,i,j)`: `True` se existe no grafo `g` uma aresta entre os nós `i` e `j`, e `False` caso contrário;
- `delnode(g,i)`: grafo que resulta de `g` eliminando o nó `i` e todas as arestas que o envolvem;
- `dim(g)`: número de nós do grafo `g`.

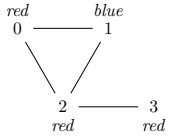
Pretende-se, em *Python*, representar cada grafo como uma matriz quadrada `m` cujo número de linhas e colunas é o número de nós do grafo e em que cada entrada `m[i][j]` é `1` se existe uma aresta a ligar os nós `i` e `j`, e `0` caso contrário. Nomeadamente, `[[0,1,1,0],[1,0,1,0],[1,1,0,1],[0,0,1,0]]` deverá ser uma representação do grafo



a) Apresente implementações eficientes apenas para as operações `edge` e `delnode`.

(vsff)

b) Uma coloração de um grafo é uma atribuição de cores aos nós do grafo, como exemplificado.



Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `monopathQ` que recebendo um grafo `g`, a lista de cores que desejamos atribuir aos nós do grafo, por ordem, e uma lista de nós $[\mathtt{n}_1, \ldots, \mathtt{n}_t]$ devolve `True` se todos os nós da lista têm a mesma cor e formam um caminho no grafo, i.e., há uma aresta a ligar cada nó $\mathtt{n}_a$ ao nó seguinte $\mathtt{n}_{a+1}$, e `False` caso contrário.

Nomeadamente, para o grafo acima colorido com a respectiva lista de cores [”`red`”, ”`blue`”, ”`red`”, ”`red`”] e o caminho $[0, 2, 3, 2, 3]$ a resposta deverá ser `True`.

IST - LMAC, MEBiom

Elementos de Programação

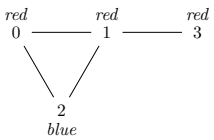
15 de Dezembro de 2017 Ficha 3D Duração: 15m

Número: _____ **Nome:** _____

Considere grafos (não-dirigidos) coloridos, onde se assume que os nós são identificados por $0, \ldots, k-1$ para um grafo com `k` nós, com as operações:

- `nodes(w)`: dada uma lista de cores `w`, constrói o grafo colorido com `len(w)` nós, sem arestas, onde cada nó `i` tem a cor `w[i]`;
- `edge(g,i,j)`: adiciona ao grafo colorido `g` uma aresta entre os nós `i` e `j`;
- `edgeQ(g,i,j)`: `True` se existe no grafo colorido `g` uma aresta entre os nós `i` e `j`, e `False` caso contrário;
- `delnode(g,i)`: grafo colorido que resulta de `g` eliminando o nó `i` e todas as arestas que o envolvem;
- `colorofnode(g,i)`: cor associada ao nó `i` no grafo colorido `g`;
- `dim(g)`: número de nós do grafo colorido `g`.

Pretende-se, em *Python*, representar cada grafo colorido como um par $(\texttt{w}, \texttt{adj})$ onde `w` é a lista de cores associadas aos nós do grafo e `adj` é uma lista de listas da forma $[\texttt{adj}_0, \texttt{adj}_1, \ldots, \texttt{adj}_{k-1}]$ onde `k` é o número de nós do grafo e cada $\texttt{adj}_i$ é a lista dos nós que partilham uma aresta com o nó `i`. Nomeadamente, (["red", "red", "blue", "red"], [[1, 2], [0, 2, 3], [0, 1], [1]]) deverá ser uma representação do grafo



a) Apresente implementações eficientes apenas para as operações `delnode` e `colorofnode`.

(vsff)

b) Desenvolva, sobre a camada de abstracção obtida acima e assegurando a independência da implementação, uma função `multipathQ` que recebendo um grafo colorido `g` e uma lista de nós $[\mathtt{n}_1, \ldots, \mathtt{n}_t]$ devolve `True` se os nós da lista não têm todos a mesma cor e formam um caminho no grafo, i.e., há uma aresta a ligar cada nó $\mathtt{n}_a$ ao nó seguinte $\mathtt{n}_{a+1}$, e `False` caso contrário.

Nomeadamente, para o grafo colorido da página anterior e o caminho $[0, 1, 3, 1, 2]$ a resposta deverá ser `True`.