

Elementos de Programação

Novembro 2020

Mini-teste 3

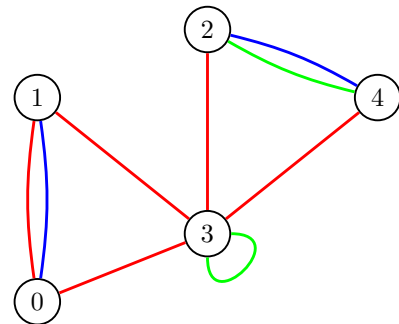
Duração: 30m

Considere *grafos não-dirigidos com arestas coloridas*, onde se assume que os nós do grafo são numerados $(0, 1, \dots, n-1)$ onde n é o número de nós do grafo, e também as possíveis cores são numeradas $(0, 1, \dots, c-1)$ onde c é o número de cores considerado para as arestas do grafo). Identificaram-se as seguintes operações:

- `grvazio(n,c)`: grafo com n nós e sem qualquer aresta (mas com c cores para colorir potenciais arestas);
- `jaresta(g,i,j,k)`: grafo que resulta de juntar ao grafo g uma aresta de cor k a ligar os nós i e j (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par (n,c) onde n é o número de nós e c o número de cores potenciais das arestas do grafo g ;
- `arestaQ(g,i,j,k)`: `True` se existe no grafo g uma aresta de cor k a ligar os nós i e j , `False` caso contrário;
- `cliqueQ(g,w,k)`: `True` se no grafo g existem arestas de cor k a ligar todos os nós distintos da lista de nós w , `False` caso contrário; por exemplo, no grafo abaixo, a lista de nós $[0, 1, 3]$ forma um clique de cor vermelha.

Em *Python*, pretende-se representar um grafo não-dirigido com arestas coloridas como um par da forma $(n, [arestascor0, arestascor1, \dots])$ onde n é o número de nós do grafo, e cada `arestascori` é a lista de arestas de cor i do grafo, representando-se cada aresta como um par de nós (o,d) com $o \leq d$.

Nomeadamente, assumindo que às três cores azul, vermelho, verde correspondem os valores 0, 1, 2, respectivamente, o grafo da ilustração é representado por $(5, [[(0, 1), (2, 4)], [(0, 1), (0, 3), (1, 3), (2, 3), (3, 4)], [(3, 3), (2, 4)]])$.



- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo não-dirigido com arestas coloridas, `False` caso contrário.

Sugestão: pode usar sem a definir uma função Booleana `listadelistasQ(e,p)` que determina se a expressão e é ou não uma lista de listas em que todos os valores satisfazem o predicado p .

Elementos de Programação

Novembro 2020

Mini-teste 3

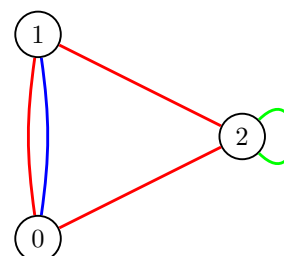
Duração: 30m

Considere *grafos não-dirigidos com arestas coloridas*, onde se assume que os nós do grafo são numerados $(0, 1, \dots, n-1)$ onde n é o número de nós do grafo), e também as possíveis cores são numeradas $(0, 1, \dots, c-1)$ onde c é o número de cores considerado para as arestas do grafo). Identificaram-se as seguintes operações:

- `grvazio(n,c)`: grafo com n nós e sem qualquer aresta (mas com c cores para colorir potenciais arestas);
- `jaresta(g,i,j,k)`: grafo que resulta de juntar ao grafo g uma aresta de cor k a ligar os nós i e j (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par (n,c) onde n é o número de nós e c o número de cores potenciais das arestas do grafo g ;
- `posscores(g,i,j)`: lista de cores das arestas de g que ligam os nós i e j ;
- `conectadoQ(g,i,k)`: `True` se no grafo g existe alguma aresta de cor k a ligar o nó i com algum outro nó, `False` caso contrário; por exemplo, no grafo abaixo, o nó 2 só é conectado na cor vermelha.

Em *Python*, pretende-se representar um grafo não-dirigido com arestas coloridas como um par da forma (c,m) onde c é o número de potenciais cores das arestas, e m é uma matriz simétrica $n \times n$ (onde n é o número de nós do grafo) em que cada entrada $m[i][j]$ é a lista das cores das arestas do grafo que ligam os nós i e j .

Nomeadamente, assumindo que às três cores azul, vermelho, verde correspondem os valores 0, 1, 2, respectivamente, o grafo da ilustração é representado por $(3, [[[], [0, 1], [1]], [[0, 1], [], [1]], [[1], [1], [2]]])$.



- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo não-dirigido com arestas coloridas, `False` caso contrário.

Sugestão: pode usar sem a definir uma função Booleana `matrizsimQ(e,p)` que determina se a expressão e é ou não uma matriz simétrica em que todas as entradas satisfazem o predicado p .

Elementos de Programação

Novembro 2020

Mini-teste 3

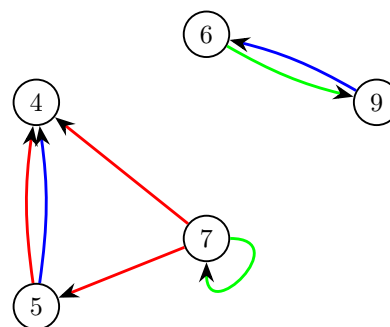
Duração: 30m

Considere *grafos dirigidos com arestas coloridas*, onde se assume que as possíveis cores das arestas são numeradas $(0, 1, \dots, c-1)$ onde c é o número de cores considerado para as arestas do grafo). Identificaram-se as seguintes operações:

- `grvazio(w,c)`: grafo cujos nós são os elementos da lista w , e sem qualquer aresta (mas com c cores para colorir potenciais arestas);
- `jaresta(g,o,d,k)`: grafo que resulta de juntar ao grafo g uma aresta de cor k do nó o para o nó d (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par (w,c) onde w é a lista de nós e c o número de cores potenciais das arestas do grafo g ;
- `posscores(g,o,d)`: lista de cores das arestas do nó o para o nó d no grafo g ;
- `origemQ(g,o,k)`: `True` se no grafo g existe alguma aresta de cor k a partir do nó o , `False` caso contrário; por exemplo, no grafo abaixo, o nó 6 só é origem de uma seta de cor verde.

Em *Python*, pretende-se representar um grafo dirigido com arestas coloridas como um triplo da forma $(w,c,arestas)$ onde w é a lista de nós do grafo, c é o número de possíveis cores das arestas do grafo, e $arestas$ é a lista das arestas do grafo, representando-se cada aresta como um triplo (o,d,k) onde o,d são os nós de origem e destino da aresta e k a sua cor.

Nomeadamente, assumindo que às três cores azul, vermelho, verde correspondem os valores 0, 1, 2, respectivamente, o grafo da ilustração é representado por $([4,5,6,7,9], 3, [(5,4,0), (5,4,1), (7,4,1), (7,5,1), (7,7,2), (6,9,2), (9,6,0)])$.



- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo dirigido com arestas coloridas, `False` caso contrário.

Sugestão: pode usar sem a definir uma função Booleana `listaQ(e,p)` que determina se a expressão e é ou não uma lista em que todos os valores satisfazem o predicado p .

Elementos de Programação

Novembro 2020

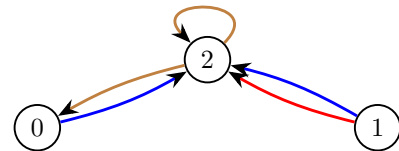
Mini-teste 3

Duração: 30m

Considere *grafos dirigidos com arestas coloridas*, onde se assume que os nós do grafo são numerados $(0, 1, \dots, n-1)$ onde n é o número de nós do grafo). Identificaram-se as seguintes operações:

- `grvazio(n,w)`: grafo com n nós e sem qualquer aresta (mas com as cores da lista w para colorir potenciais arestas);
- `jaresta(g,o,d,k)`: grafo que resulta de juntar ao grafo g uma aresta de cor k do nó o para o nó d (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par (n,w) onde n é o número de nós e w a lista de cores potenciais das arestas do grafo g ;
- `arestaQ(g,o,d,k)`: `True` se no grafo g existe uma aresta de cor k do nó o para o nó d , `False` caso contrário;
- `completoQ(g,wnos)`: `True` se no grafo g existem arestas em ambas as direções entre quaisquer dois nós distintos da lista $wnos$, `False` caso contrário; por exemplo, no grafo abaixo, a lista de nós $[0,2]$ é completa.

Em *Python*, pretende-se representar um grafo dirigido com arestas coloridas como um par da forma (w,m) onde w é a lista de possíveis cores das arestas do grafo, e m é uma matriz $n \times n$ (em que n é o número de nós do grafo) onde cada entrada $m[i][j]$ é a lista das cores das arestas do nó i para o nó j no grafo.



Nomeadamente, o grafo da ilustração é representado por $([\text{blue}, \text{red}, \text{brown}], [[[], [], [\text{blue}]], [[], [], [\text{red}, \text{blue}]], [[\text{brown}], [], [\text{brown}]]])$.

- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo dirigido com arestas coloridas, `False` caso contrário.

Sugestão: pode usar sem a definir uma função Booleana `matrizquadQ(e,p)` que determina se a expressão e é ou não uma matriz quadrada em que todas as entradas satisfazem o predicado p .

Elementos de Programação

Novembro 2020

Mini-teste 3

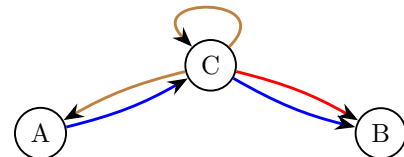
Duração: 30m

V5

Considere *grafos dirigidos com arestas coloridas*. Identificaram-se as seguintes operações:

- `grvazio(wnos, wcores)`: grafo cujos nós são os elementos da lista `wnos`, e sem qualquer aresta (mas com as cores da lista `wcores` para colorir potenciais arestas);
- `jaresta(g, o, d, k)`: grafo que resulta de juntar ao grafo `g` uma aresta de cor `k` do nó `o` para o nó `d` (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par `(wnos, wcores)` onde `wnos` é a lista de nós e `wcores` a lista de cores potenciais das arestas do grafo `g`;
- `posscores(g, o, d)`: lista das cores das arestas do nó `o` para o nó `d` no grafo `g`;
- `orientadoQ(g)`: `True` se no grafo `g` não existe nenhum par de nós distintos acessíveis por arestas em ambas as direções, `False` caso contrário; por exemplo, o grafo abaixo não é orientado devido às arestas entre os nós A e C.

Em *Python*, pretende-se representar um grafo dirigido com arestas coloridas como um triplo da forma `(wnos, wcores, m)` onde `wnos` é a lista de nós de grafo, `wcores` a lista de possíveis cores das arestas do grafo, e `m` é uma matriz `len(wnos) × len(wcores)` onde cada entrada `m[i][j]` é a lista dos nós destino das arestas do grafo com origem no nó `wnos[i]` e cor `wcores[j]`.



Nomeadamente, o grafo da ilustração é representado por `([A, B, C], [blue, red, brown], [[C], [], []], [[], [], []], [[B], [B], [A, C]])`.

- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo dirigido com arestas coloridas, `False` caso contrário.

Sugestão: pode usar sem a definir uma função Booleana `matrizdimQ(e, lin, col, p)` que determina se a expressão `e` é ou não uma matriz com `lin` linhas e `col` colunas em que todas as entradas satisfazem o predicado `p`.

Elementos de Programação

Novembro 2020

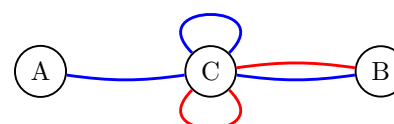
Mini-teste 3

Duração: 30m

Considere *grafos não-dirigidos com arestas coloridas*. Identificaram-se as seguintes operações:

- `grvazio(wnos, wcores)`: grafo cujos nós são os elementos da lista `wnos`, e sem qualquer aresta (mas com as cores da lista `wcores` para colorir potenciais arestas);
- `jaresta(g, i, j, k)`: grafo que resulta de juntar ao grafo `g` uma aresta de cor `k` entre os nós `i` e `j` (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par `(wnos, wcores)` onde `wnos` é a lista de nós e `wcores` a lista de cores potenciais das arestas do grafo `g`;
- `arestaQ(g, i, j, k)`: `True` se existe no grafo `g` uma aresta de cor `k` entre os nós `i` e `j`, `False` caso contrário;
- `rainbowQ(g)`: `True` se qualquer que seja a aresta do grafo `g` existem arestas entre os mesmos nós com todas as cores possíveis, `False` caso contrário; por exemplo, o grafo de duas cores abaixo não é um arco-íris pois falta-lhe uma aresta vermelha entre os nós A e C.

Em *Python*, pretende-se representar um grafo não-dirigido com arestas coloridas como um triplo da forma `(wnos, wcores, arestas)` onde `wnos` é a lista de nós de grafo, `wcores` a lista de possíveis cores das arestas do grafo, e `arestas` é a lista das arestas do grafo, onde uma aresta de cor `k` entre os nós `o` e `d` surge na lista nas duas formas `(o, d, k)` e `(d, o, k)`.



Nomeadamente, o grafo da ilustração é representado por `([A, B, C], [blue, red], [(A, C, blue), (C, A, blue), (C, C, blue), (C, C, red), (B, C, blue), (C, B, blue), (B, C, red), (C, B, red)])`.

- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo não-dirigido com arestas coloridas, `False` caso contrário.
- Sugestão: pode usar sem a definir uma função Booleana `listaQ(e, p)` que determina se a expressão `e` é ou não uma lista em que todos os valores satisfazem o predicado `p`.

Elementos de Programação

Novembro 2020

Mini-teste 3

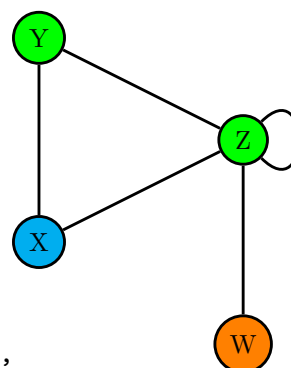
Duração: 30m

Considere *grafos não-dirigidos com nós coloridos*. Identificaram-se as seguintes operações:

- `grvazio(w)`: grafo sem qualquer aresta, cujos nós e respectiva cor correspondem aos pares (no, cor) da lista w ;
- `jaresta(g, i, j)`: grafo que resulta de juntar ao grafo g uma aresta a ligar os nós i e j (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par $(wnos, wcores)$ onde $wnos$ é a lista de nós e $wcores$ a lista de cores usados no grafo g ;
- `arestaQ(g, i, j)`: `True` se existe no grafo g uma aresta entre os nós i e j , `False` caso contrário;
- `nosdecor(g, k)`: lista de todos os nós do grafo g cuja cor é k .

Em *Python*, pretende-se representar um grafo não-dirigido com nós coloridos como um triplo da forma $(wnos, nosporcor, m)$ onde $wnos$ é a lista de nós do grafo, $nosporcor$ é uma lista de pares da forma $(k, nosk)$ com k uma cor e $nosk$ a lista dos nós de cor k do grafo, e m é uma matriz simétrica $\text{len}(wnos) \times \text{len}(wnos)$ onde cada entrada $m[i][j]$ é 1 se existe uma aresta entre os nós $wnos[i]$ e $wnos[j]$ e $m[i][j]$ é 0 se não existe.

Nomeadamente, o grafo da ilustração é representado por $([X, Y, Z, W], [(blue, [X]), (green, [Y, Z]), (orange, [W])], [[0, 1, 1, 0], [1, 0, 1, 0], [1, 1, 1, 1], [0, 0, 1, 0]])$, e poderia ser obtido juntando arestas ao grafo `grvazio([(X, blue), (Y, green), (Z, green), (W, orange)])`.



- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo não-dirigido com arestas coloridas, `False` caso contrário.

Sugestão: pode usar sem as definir funções Booleanas `listaQ(e, p)` que determina se a expressão e é ou não uma lista em que todos os valores satisfazem o predicado p , e `matrizsimdimQ(e, n, p)` que determina se a expressão e é ou não uma matriz simétrica com n linhas e colunas em que todas as entradas satisfazem o predicado p .

Elementos de Programação

Novembro 2020

Mini-teste 3

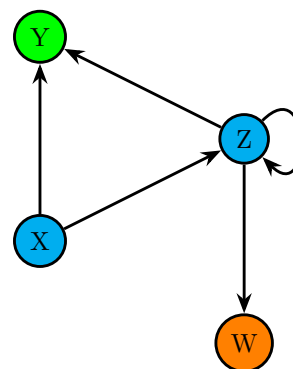
Duração: 30m

Considere *grafos dirigidos com nós coloridos*. Identificaram-se as seguintes operações:

- `grvazio(w)`: grafo sem qualquer aresta, cujos nós e respectiva cor correspondem aos pares (no, cor) da lista w ;
- `jaresta(g,o,d)`: grafo que resulta de juntar ao grafo g uma aresta do nó o para o nó d (pode assumir-se que tal aresta não existe);
- `noscores(g)`: par $(wnos, wcores)$ onde $wnos$ é a lista de nós e $wcores$ a lista de cores usadas no grafo g ;
- `destinos(g,o)`: lista dos nós destino das arestas do grafo g a partir do nó o ;
- `cordeno(g,o)`: cor do nó o do grafo g .

Em *Python*, pretende-se representar um grafo dirigido com nós coloridos como um par da forma $(w, arestas)$ onde w é a lista de pares (no, cor) do grafo, e $arestas$ é a lista das arestas do grafo na forma (o, d) para uma aresta do nó o para o nó d .

Nomeadamente, o grafo da ilustração é representado por $([(X, blue), (Y, green), (Z, blue), (W, orange)], [(X, Y), (X, Z), (Z, Y), (Z, Z), (Z, W)])$.



- (a) Apresente implementações eficientes para todas as operações.
- (b) Defina uma função `grafok` que recebendo como argumento qualquer expressão *Python* devolve `True` se a expressão corresponde à representação de um grafo não-dirigido com arestas coloridas, `False` caso contrário.
- Sugestão: pode usar sem a definir uma função Booleana `listaQ(e,p)` que determina se a expressão e é ou não uma lista em que todos os valores satisfazem o predicado p .