MAC328 - Algoritmos em grafos
Tarefa 3
Entrega: 11/9/2023, no e-disciplinas

Uma *coloração de arestas* de um grafo é uma função das arestas em algum conjunto finito (cujos elementos são chamados tradicionalmente de cores) tal que todas as arestas incidentes a cada vértice tenham cores distintas. Equivalentemente, é uma partiçao do conjunto de arestas em emparelhamentos (que não precisam ser perfeitos). O número mínimo de cores necessário para colorir as arestas de um grafo G é chamado de *índice cromático* de G e denotado $\chi'(G)$.

É fácil ver que $\chi'(G) \geq \Delta(G)$, onde essa notação indica o grau máximo de (vértices de) G. Afinal, existe um vértice de grau $\Delta(G)$, e é preciso colorir cada aresta incidente a ele com uma cor diferente. Essa é a parte fácil do TEOREMA DE VIZING (1964):

Para todo grafo G, $\Delta(G) \leq \chi'(G) \leq \Delta(G)$.

Como sempre, a Wikipedia traz um bom artigo sobre o assunto, Edge coloring.

Uma coloração de arestas é dita *acíclica* se nenhum ciclo é bicolorido. Existe uma conjectura de que bastam $\Delta(G) + 2$ cores para tal coloração em um grafo, mas ela está longe de ter sido provada. O artigo na Wikipedia menciona isso brevemente.

No artigo *Esperet L, Parreau A,* Acyclic edge-coloring using entropy compression. Eur J Comb 34(6):1019-1027

os autores mostram que $4(\Delta(G) - 1)$ cores são suficentes.

Questões:

A questões 1 e 2 devem ser entregues num arquivo pdf produzido a partir do LaTeX. O nome do arquivo deve ser `tarefa3-nusp.pdf`, onde `nusp` é o seu número USP. No topo da página devem constar seu nome e número USP de forma clara.

1. Um grafo é *regular* se todos os vértices tem o mesmo grau; se este grau é r, o grafo é r-*regular*. Mostre que todo grafo bipartido r-regular admite uma coloração de arestas com r cores.

2. Esta questão é facultativa. Mostre um grafo r-regular, para algum $r \geq 3$, que não pode ser aresta-colorido com 3 cores (prove isso). Claro que esse grafo não pode ser bipartido.

3. Implemente o algoritmo de Esperet e Parreau no SAGE, e entregue num arquivo separado.

   (a) O nome do arquivo deve ser `tarefa3-nusp.sage`, onde `nusp` é o seu número USP.

   (b) Na primeira linha do arquivo deve haver uma linha de comentário, contendo seu nome e número USP

   (c) O arquivo deve definir uma função `colore(G)`, que recebe um grafo G e devolve uma cópia de G com as arestas rotuladas por inteiros positivos, de tal forma que essa rotulação seja uma coloração acíclica.

   (d) Como sempre, não há restrições em como estruturar seu codigo, nem em usar seja lá o que for que SAGE proporciona, mas se espera um código compreensível por seres humanos.

Segue aqui o algoritmo citado, em pseudo código, e com uma pequena simplificação:

COLORE(G)
    Comece com as arestas de G descoloridas
    Ordene $G.E$ arbitrariamente
    Cores $= \{1, 2, \ldots, 4(\Delta(G) - 4)\}$
    $K = 2(\Delta(G) - 1)$
    while houver aresta não colorida
        $e =$ menor aresta não colorida
        $S =$ Cores $-$ {cores em arestas incidentes a alguma ponta de e}
        sorteie um inteiro f entre 1 e K e atribua a e a f-ésima cor em S
        if existe ciclo bicolorido contendo a aresta e
            Seja $e_0 = e, e_1, e_2, \ldots, e_k$ um tal circuito
            Descolora $e_0, e_1, e_2, \ldots, e_{k-2}$ (as duas últimas mantêm as cores)
    return a coloração

Observações:

(a) Para o **sorteie**, use no sage `randint` ou `randrange`, conforme sua conveniência.

(b) Este é um algoritmo probabilístico, com um pouco de backtracking. Os autores provam que tem probabilidade positiva de parar depois de muitíssimas iterações. Na prática, para bem rápido.

(c) Note que a instrução de sorteio supões que S tem pelo menos K elementos. Isso é fácil de provar.

(d) As funções `mostra` e `mostrac` em `graph-utils` desenham o grafo colorido com cores de verdade.

(e) O utilitário `test-coloring` testa a rotulação nas arestas de um grafo é uma coloração acíclica.