UFMG/ICEx/DCC — DCC206 – Algoritmos 1

Trabalho Prático 1
Grafos Planares

Ciências Exatas & Engenharias — 1º Semestre 2024

Observações:

1. Comece a fazer este trabalho imediatamente. Você nunca terá tanto tempo para resolvê-lo quanto agora!
2. **Data de entrega**: 18 de abril de 2024, até **às 23:59 horas**, ou antes.
3. **Submissão**: Faça a submissão deste trabalho no Moodle, conforme instruções postadas lá.
4. **Plataforma computacional**: O seu trabalho será executado na plataforma VPL do Moodle.
5. **Linguagem**: Você deve escrever o seu programa obrigatoriamente na linguagem de programação C++. Não será aceita outra linguagem.
6. **Documentação**: Veja instruções no Moodle (aguardando a definição do monitor).
7. **Testes**: O seu programa será avaliado conforme descrito no Moodle da disciplina.

Grafos planares

Objetivo do trabalho

Neste trabalho, vamos exercitar um problema relacionado à primeira parte da disciplina: grafos e alguns algoritmos elementares em nessas estruturas. Para tal, trabalharemos com um problema em uma classe de grafos conhecida como grafos planares.

Informações importantes. Veja o Moodle da disciplina para informações sobre a submissão.

Definição do problema

Definição. Um grafo planar é um grafo que pode ser desenhado em um plano sem que suas arestas se cruzem. Ou seja, é um grafo que pode ser representado por um diagrama bidimensional sem sobreposições.

História. Alguns fatos históricos:

- Euler (1736) O matemático Leonhard Euler resolveu o problema das sete pontes de Königsberg, que é um problema de grafos planares. Este foi um dos primeiros estudos formais sobre grafos planares.
- Kuratowski (1930) O matemático Kazimierz Kuratowski provou o Teorema de Kuratowski, que caracteriza grafos planares como aqueles que não contêm subgrafos homeomórficos ao grafo completo K5 ou ao grafo bipartido completo K3,3. Este teorema fornece uma condição necessária e suficiente para que um grafo seja planar.
- Fáry (1948) O matemático István Fáry provou o Teorema de Fáry, que afirma que todo grafo planar pode ser desenhado em um plano com todas as suas arestas retas. Este teorema mostra que é possível representar grafos planares de forma simples e intuitiva.
- Tutte (1950) O matemático William Tutte provou o Teorema de Tutte, que fornece uma caracterização algébrica de grafos planares. Este teorema é importante para o estudo teórico de grafos planares.

Aplicações. Os grafos planares são usados ??em diversas áreas, como:

- Cartografia – para representar mapas de países e regiões.
- Circuitos elétricos – para analisar redes elétricas.
- Redes de computadores – para modelar redes de comunicação.
- Redes de linhas de transmissão de energia – para modelar essas redes.
- Linhas de produção de uma fábrica – para modelar as linhas de produtos.
- Geometria computacional – para resolver problemas de geometria algorítmica.

Alguns problemas importantes relacionados a grafos planares são:

- Problema de coloração de grafos – determinar o menor número de cores necessárias para colorir um grafo planar de modo que vértices adjacentes tenham cores diferentes.
- Problema do isomorfismo de grafos planares – determinar se dois grafos planares são isomórficos, ou seja, se eles podem ser transformados um no outro por uma deformação contínua.
- Problema do empacotamento de grafos – determinar como colocar o maior número possível de grafos planares em um determinado espaço.

O estudo de grafos planares tem uma longa história e está relacionado a importantes problemas matemáticos e práticos. Um deles, é o problema de coloração, descrito acima.

O Problema das Quatro Cores (P4C)

Esse é um dos problemas mais famosos da Matemática e da Ciência da Computação. A questão é simples: qual é o menor número de cores necessárias para colorir um mapa de modo que países adjacentes tenham cores diferentes?

Em 1852, Francis Guthrie conjecturou que quatro cores seriam suficientes para qualquer mapa. A conjectura permaneceu sem prova por mais de um século, até que em 1976, Kenneth Appel e Wolfgang Haken finalmente a provaram usando um método controverso de prova assistida por computador.

A busca por uma prova mais elegante e compreensível do P4C impulsionou o desenvolvimento da teoria dos grafos. O P4C pode ser reformulado como um problema de coloração de grafos, onde cada região é representada por um vértice e cada fronteira comum por uma aresta.

Alguns marcos importantes na história do P4C:

- 1852 Francis Guthrie propõe a conjectura das quatro cores.
- 1879 Alfred Kempe publica uma prova que posteriormente é refutada.
- 1930s Hassler Whitney e Philip Hall introduzem conceitos importantes da teoria dos grafos que são usados ??para estudar o P4C.
- 1976 Kenneth Appel e Wolfgang Haken publicam a primeira prova do P4C usando um método computacional.
- 1980s Neil Robertson, Paul Seymour e Robin Thomas desenvolvem a teoria dos grafos menores, que fornece uma estrutura para analisar o P4C e outros problemas de coloração de grafos.
- 2005 Uma equipe liderada por Georges Gonthier publica uma prova formalizada do P4C usando o assistente de prova Coq.

O P4C continua a ser um tema de pesquisa ativo, com matemáticos buscando novas e mais elegantes provas, explorando generalizações e aplicações em outras áreas da Matemática e da ciência da computação.

Algumas curiosidades:

- O P4C foi um dos primeiros problemas a ser solucionado com a ajuda de um computador.
- A prova de Appel e Haken foi inicialmente controversa por causa do uso extensivo de computação.
- A teoria dos grafos desenvolvida para estudar o P4C tem aplicações em diversas áreas, como redes sociais, logística e criptografia.

O P4C é um problema clássico com uma história rica e um impacto significativo na Matemática e na ciência da computação. A busca por sua solução impulsionou o desenvolvimento de importantes ferramentas matemáticas e inspirou pesquisas em diversas áreas.

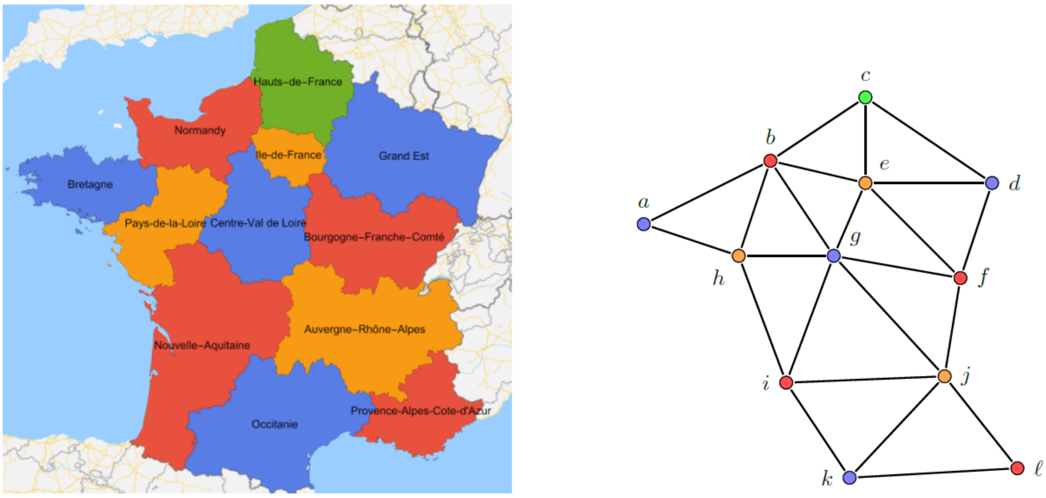
Veja, por exemplo, o mapa da França na Figura 1:

Depois de muitas pessoas diferentes colorirem mapas, elas perceberam que nunca precisaram de mais de quatro cores e durante mais de um século não sabíamos se existia um mapa que precisava de uma quinta cor. Esse problema só foi resolvido no fim do século XX e hoje é conhecido como teorema das quatro cores.

Este trabalho não tem (quase) nenhuma relação com o teorema em si, mas esse último motivou o estudo de grafos muito importantes hoje em dia: os grafos planares. Um grafo é dito planar se existe uma forma de desenhar seus vértices e arestas numa folha de papel sem que arestas se cruzem. Esses grafos têm propriedades fundamentais para a construção de algoritmos eficientes; a mais interessante delass para nós neste momento é a existência do que chamamos de faces: regiões delimitadas por arestas que não são inteiramente subdividas por nenhuma outra aresta do grafo. Podemos pensar nas faces como regiões diferentes do plano e os vértices/arestas que delimitam a face estão na região chamada borda da face, que é o tema principal deste trabalho. Por exemplo, o grafo da Figura 2 possui cinco faces.

Note que F_1 é composta sempre pelos vértices de fora do grafo e por isso é chamada de face externa do grafo. Com isso, temos que cada face é uma sequência de vértices onde o primeiro e o último são iguais e dois vértices consecutivos são adjacentes. Ou seja, podemos descrever as faces do grafo acima como: $F_1 = \langle b, e, f, c, g, b \rangle$, $F_2 = \langle c, g, h, g, b, a, c \rangle$, $F_3 = \langle a, b, d, c, a \rangle$, $F_4 = \langle e, c, d, b, e \rangle$, e $F_5 = \langle f, c, e, f \rangle$. Repare também que um grafo planar pode ser desenhado de diferentes maneiras no plano e a forma de desenho **interfere nos vértices que compõem cada face do grafo**; por exemplo, no grafo da Figura 2, podemos mover h para a esquerda de G, fazendo com que h participe da face externa mas deixe de participar da face F_2.

Neste trabalho, dado um grafo planar G e as coordenadas de seus vértices em um desenho válido de G (ou seja, sem cruzamento de arestas e todas as arestas sendo segmentos de reta entre os pontos de seus extremos), você deve listar todas as faces desse grafo.



(a) (b)

Figura 1: (a) Mapa da França onde cada região tem uma cor diferente de todas as suas vizinhas; (b) grafo que representa o mapa ‘a esquerda com as cores apropriadas.

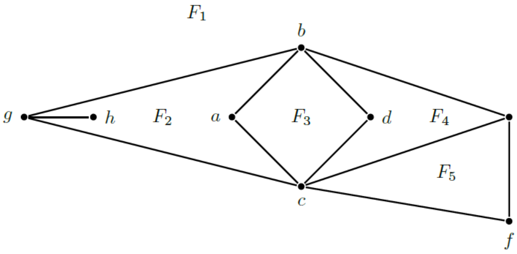


Figura 2: Um grafo e suas cinco faces F_1, F_2, F_3, F_4, F_5.

Dicas

Existem várias soluções possíveis para esse problema, algumas bem complicadas e outras (esperadas) mais simples. Todas elas exigem um pouquinho de geometria computacional, por isso abaixo tem algumas funções que podem ser úteis.

```
struct Ponto {
  double x, y;
};

// Distância euclidiana de a para b.
double Distancia(Ponto a, Ponto b) {
  double x = (a.x - b.x), y = (a.y - b.y);
  return sqrt(x*x + y*y);
}

// Coeficiente da reta que passa na origem e p.
double Inclinação(Ponto p) {
  return atan2(p.y, p.x);
}

// Coeficiente da reta orientada de p para q.
double InclinaçãoRelativa(Ponto p, Ponto q) {
  return atan2(q.y - p.y, q.x - p.x);
}

// Determina se ao caminhar de a para b e depois de b para c
estamos fazendo uma curva ‘a esquerda, ‘a direita, ou seguindo em frente.
int TipoCurva(Ponto a, Ponto b, Ponto c) {
    double v = a.x*(b.y-c.y)+b.x*(c.y-a.y)+c.x*(a.y-b.y);
    if (v < 0) return -1; // esquerda.
    if (v > 0) return +1; // direita.
    return 0; // em frente.
}
```

Questões. Abaixo vão algumas perguntas relevantes para você se fazer antes de começar a programar:

Pergunta 1. A quantas faces uma dada aresta pode pertencer?

Pergunta 2. Dado que acabei de atravessar uma aresta e de u para v, existe alguma outra aresta incidente v que com certeza está em uma mesma face que e?

Pergunta 3. Como posso evitar de passar por uma mesma face mais de uma vez?

Para responder a essas perguntas, olhe novamente para a Figura 1 e tente pensar em como podemos gerar todas as faces que contém o vértice a se a primeira aresta que pegarmos na hipótese da questão 2 for incidente a a.

Pergunta 4. Será que se fizer alguma transformação nas arestas minha vida fica mais fácil?

Dica da dica: pense em orientações/curvas para essa última pergunta.

Casos de teste

Formatado da entrada. Cada caso de teste é composto por várias linhas. A primeira linha contém dois inteiros, N e M, que representam, respectivamente, o número de vértices e arestas do grafo de entrada G. É garantido que G é conexo, que $1 \leq N, M \leq 10^5$, e que $V(G) = \{1, 2, \ldots, N\}$. A i-ésima linha da entrada começa com dois números reais x_i, y_i, que representam as coordenadas do vértice i no plano cartesiano; é garantido que $-10^4 \leq x_i, y_i \leq 10^4$. Na mesma linha, temos um inteiro positivo d_i, que representa o grau do vértice i. Ainda na mesma linha, temos d_i inteiros entre 1 e N, cada um correspondendo a um vizinho de i; é garantido que um vértice não é vizinho de si mesmo.

Formatado da saída. A primeira linha da saída contém um inteiro F, que deve ser igual ao número de faces de G. As próximas F linhas devem corresponder às F faces de G; cada linha começa com um inteiro s_i, que representa o tamanho da i-ésima face de G; em seguida, existem si inteiros, representando o circuito que corresponde à borda da face. A ordem das faces não é importante, mas vértices consecutivos na borda da face devem ser adjacentes.

Limites de execução. Para qualquer caso de teste, seu código deve imprimir a resposta em no máximo 3 segundos. Seu programa deve usar menos de 100MB de memória. Estruturas de dados devem ser alocadas sob demanda; ou seja, não faça vetores estáticos gigantescos para grafos com poucos vértices. Todas as avaliações serão feitas automaticamente via VPL. Programas que não aderirem a essas restrições para um teste terão a nota do mesmo zerada.

Lembre-se: você pode submeter uma solução para a tarefa no máximo x vezes e apenas a última submissão será levada em conta para fins de avaliação; é garantido que $x \leq 10$.

Exemplos

Exemplo da Figura 2. No trecho abaixo, o vértice de número i corresponde à i-ésima letra do alfabeto;, i.e., $4 = d$.

Entrada	Saída
8 11	5
0 0 2 2 3	6 2 5 6 3 7 2
1 1 4 1 4 5 7	7 3 7 8 7 2 1 3
1 -1 5 1 4 5 6 7	5 1 2 4 3 1
2 0 2 2 3	5 5 3 4 2 5
4 0 3 2 3 6	4 6 3 5 6
4 -1.5 2 3 5	
-3 0 3 2 3 8	
-2 0 1 7	

Exemplo da Figura 1b. No trecho abaixo, o vértice de número i corresponde à i-ésima letra do alfabeto;, i.e., $4 = d$.

Entrada	Saída
12 23	13
-3 1 2 2 8	11 1 8 9 11 12 10 6 4 3 2 1
-1 2 5 1 3 5 7 8	4 1 2 8 1
0.5 3 3 2 5 4	4 3 2 5 3
2.5 1.65 3 3 5 6	4 2 8 7 2
0.5 1.65 5 2 3 4 6 7	4 2 5 7 2
2 0.15 4 4 5 7 10	4 3 4 5 3
0 0.5 6 2 5 6 8 9 10	4 5 7 6 5
-1.5 0.5 4 1 2 7 9	4 5 4 6 5
-0.75 -1.5 4 7 8 10 11	4 7 8 9 7
1.75 -1.4 5 6 7 9 11 12	4 7 9 10 7
0.25 -3 3 9 10 12	4 7 6 10 7
2.9 -2.85 2 10 11	4 9 10 11 9
	4 10 11 12 10

Exemplo bônus. A seguir, há um outro exemplo extra.

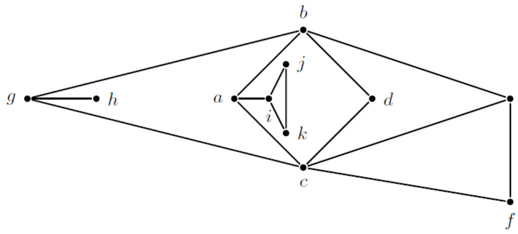


Figura 3: Exemplo bônus.

```
Entrada                          Saída
11 15                            6
0 0 3 2 3 9                      10 1 3 4 2 1 9 10 11 9 1
1 1 4 1 4 5 7                    7 1 2 7 8 7 3 1
1 -1 5 1 4 5 6 7                 6 2 5 6 3 7 2
2 0 2 2 3                        5 2 4 3 5 2
4 0 3 2 3 6                      4 3 6 5 3
4 -1.5 2 3 5                     4 9 11 10 9
-3 0 3 2 3 8
-2 0 1 7
0.5 0 3 1 10 11
0.75 0.5 2 11 9
0.75 -0.5 2 10 9
```