1 Árvores

Definição 1.1 : *Uma **árvore** é um grafo simples conexo e sem ciclos.*

Um grafo simples sem ciclos mas não conexo (em que cada componente conexa é portanto uma árvore) chama-se uma **floresta**.
Numa árvore (ou numa floresta), um vértice de grau 1 é uma **folha**.

Ao contrário do que se passa para grafos em geral, o número de arestas de uma árvore é determinado pelo número de vértices: se T é uma árvore com n vértices então tem $n-1$ arestas. Provamos de facto um pouco mais:

Proposição 1.2 : *Seja G um grafo simples com n vértices. As seguites condições são equivalentes:*

a) G é uma árvore;

b) G é conexo e tem $n-1$ arestas;

c) G tem $n-1$ arestas e não tem ciclos.

Demonstração 1.3 : *Provamos a) $\Rightarrow$ b) $\Rightarrow$ c) $\Rightarrow$ a).*
Seja G conexo e sem ciclos; provamos por indução no número de vértices n que G tem $n-1$ arestas. Os casos $n=1$ e $n=2$ são evidentes; suponhamos portanto que a propriedade vale para grafos conexos, sem ciclos, com menos que n arestas; Se eliminarmos uma aresta uv de G ficamos com um grafo G' sem ciclos e com duas componentes conexas (se houvesse um caminho em G' de u para v, então em G esse caminho adicionado da aresta uv seria um ciclo). A hipótese de indução aplica-se a cada uma das componentes: se elas têm, respectivamente, p e q vértices, terão $p-1$ e $q-1$ arestas; mas G tem os mesmos vértices e mais uma aresta que G', logo o número de arestas de

G é $p-1+q-1+1=n-1$ como queríamos demonstrar.
Suponhamos agora que G é conexo e tem $n-1$ arestas; se G tivesse ciclos, poderíamos eliminar arestas mantendo o grafo conexo (para "abrir" os ciclos); no fim desse processo teríamos uma árvore com n vértices mas menos que $n-1$ arestas, o que já vimos ser impossível.
Suponhamos finalmente que G tem $n-1$ arestas e não tem ciclos. Sejam $G_1, \cdots G_k$ as componentes conexas de G e seja v_i o número vértices de G_i; cada G_i é uma árvore logo tem $v_i - 1$ arestas. Mas então

$$n-1=\sum_{i=1}^{k}(v_i-1)=\sum_{i=1}^{k}v_i-k=n-k$$

e portanto $k=1$, ou seja, G é conexo.

Como consequência, a sequência de graus $d_1 \geq \cdots \geq d_n$ de uma árvore de ordem n satisfaz as condições

$$d_i > 0,\ \forall i; \qquad \sum_{i=1}^{n} d_i = 2(n-1)$$

Prova-se (por indução em n) que

Proposição 1.4 *: Dados inteiros positivos $d_1, \cdots, d_n$ existe uma árvore com vértices $v_1, \cdots, v_n$ e $d(v_i) = d_i$ se e só se*

$$\sum_{i=1}^{n} d_i = 2(n-1)$$

Nota 1.5 *Se a sequência de graus de um grafo simples G satisfaz*

$$\sum_{i=1}^{n} d_i = 2(n-1)$$

isso não implica que G seja uma árvore. No entanto, se G for conexo aquela condição implica que se trata de uma árvore.

Outras caracterizações e propriedades de árvores deduzem-se facilmente e várias delas são deixadas como exercícios.

Exercícios X.1

1. Dada uma árvore de ordem $n > 1$, mostrar que o número de folhas (vértices de grau 1) é igual a
$$2 + \sum_{d(v) \geq 2} (d(v) - 2)$$
em que a soma se faz sobre todos os vértices com grau maior ou igual a 2.

2. Seja T uma árvore com n vértices, tendo exactamente quatro deles grau 4, cinco grau 3, e todos os outros grau 1. Determinar o valor de n.

3. Sendo $m = \dfrac{\sum d(v)}{|V|}$ a média dos graus dos vértices da árvore $\mathbf{T}$, calcular o número de vértices $|V|$ em função de m.

4. Uma floresta F tem n vértices e m arestas. Quantas árvores (componentes conexas) tem F?

5. Mostrar que as seguintes condições são equivalentes:
   1) G é uma árvore;
   2) para cada par de vértices u, v de G existe um único caminho em G de u para v.
   3) G não tem ciclos mas se acrescentarmos uma aresta a, G tem exactamente um ciclo.

6. Mostrar que uma sequência
$$d_1 \geq d_2 \geq \cdots \geq d_n \geq 1$$
é a sequência ordenada de graus de uma árvore com $n > 1$ vértices $v_1, \cdots, v_n$, se e só se
$$\sum_{i=1}^{n} d_i = 2(n-1).$$

7. Dada uma árvore T com pelo menos 3 vértices, seja T' a árvore que se obtém eliminando todas as folhas de T. Mostrar que T e T' têm o mesmo centro.
   Concluir que o centro de T consiste em um único vértice ou em dois vértices adjacentes.

1.1 Árvores Geradoras

Definição 1.6 *:Uma **árvore geradora** de um grafo G é um subgrafo de G que contenha todos os vértices de G e seja uma árvore.*

Naturalmente, uma condição necessária para que G tenha uma árvore geradora é que seja conexo. Essa condição é também suficiente e existem algoritmos simples para a construção de uma árvore geradora.

1.1.1 Busca em largura e em profundidade

Descrevemos brevemente dois algoritmos de busca em grafos que podem ser usados para determinar uma árvore geradora. Ambos têm como ponto de partida a escolha de um vértice r (a raiz da árvore); em cada passo do algoritmo é actualizada uma lista de vértices visitados e há um vértice activo a partir do qual se continua a busca.

Busca em largura Na busca em largura a lista é iniciada com o vértice r e são adicionados sucessivamente ao fim da lista os vértices, não visitados anteriormente, adjacentes ao vértice activo, que é sempre o primeiro da lista; a cada novo vértice v adicionado à lista corresponde uma aresta da árvore incidente em v e no vértice activo; quando já não há novos vértices nessas condições, o primeiro vértice da lista é apagado e o vértice seguinte toma o lugar de vértice activo; o algoritmo termina quando a lista está vazia.

Busca em profundidade Na busca em profundiadade a diferença está em que o vértice activo em cada momento é o último da lista, que é apagado quando não é possível adicionar à lista um vértice adjacente ainda não visitado.

Em ambos os casos o algoritmo termina quando todos os vértices da componente conexa do grafo que contém o vértice r foram visitados e determina uma árvore geradora dessa componente conexa.

Exemplo 1.7 *Para exemplificar a aplicação destes algoritmos considere-se o grafo simpes com matriz de adjacência*

$$\begin{pmatrix}
0&1&1&1&1&0&0&0&0&0&0&0&0\\
1&0&0&0&0&1&1&0&0&0&0&0&0\\
1&0&0&1&0&0&1&1&1&0&0&0&0\\
1&0&1&0&0&0&1&0&0&1&0&0&0\\
1&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&0&0&1&0&0&0&1&0\\
0&1&1&1&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&1\\
0&0&0&1&1&0&0&0&0&0&1&0&1\\
0&0&0&0&1&0&0&0&1&1&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&0
\end{pmatrix}$$

Tomamos em ambos os casos o vértice 1 *como raiz e a busca é feita usando a ordem dos índices dos vértices; na busca em largura, a lista de vértices vai evoluindo do seguinte modo*

$$1 \to 1,2 \to 1,2,3 \to 1,2,3,4 \to 1,2,3,4,5 \to 2,3,4,5 \to 2,3,4,5,6 \to$$

$$\to 2,3,4,5,6,7 \to 3,4,5,6,7 \to 3,4,5,6,7,8 \to 3,4,5,6,7,8,9 \to$$

$$\to 4,5,6,7,8,9 \to 4,5,6,7,8,9,10 \to 5,6,7,8,9,10 \to$$

$$\to 5,6,7,8,9,10,11 \to 6,7,8,9,10,11 \to 6,7,8,9,10,11,12 \to$$
$$7,8,9,10,11,12 \to 8,9,10,11,12 \to 9,10,11,12 \to 9,10,11,12,13 \to$$
$$\to 10,11,12,13 \to 11,12,13, \to 12,13 \to 13 \to \emptyset$$

e a árvore tem as arestas

$\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{2,6\},\{2,7\},\{3,8\},\{3,9\},\{4,10\},\{5,11\},\{6,12\},\{9,13\}$

enquanto que na busca em profundidade temos

$$1 \to 1,2 \to 1,2,6 \to 1,2,6,8 \to 1,2,6,8,3 \to 1,2,6,8,3,4 \to$$
$$\to 1,2,6,8,3,4,10 \to 1,2,6,8,3,4,10,5 \to 1,2,6,8,3,4,10,5,11 \to$$
$$\to 1,2,6,8,3,4,10,5,11,9 \to 1,2,6,8,3,4,10,5,11,9,13 \to 1,2,6,8,3,4,10,5,11,9$$
$$\to 1,2,6,8,3,4,10,5,11 \to 1,2,6,8,3,4,10,5 \to 1,2,6,8,3,4,10$$
$$\to 1,2,6,8,3,4 \to 1,2,6,8,3 \to 1,2,6,8,3,7 \to 1,2,6,8,3,7,12$$
$$\to 1,2,6,8,3,7 \to 1,2,6,8,3 \to 1,2,6,8 \to 1,2,6 \to 1,2$$
$$\to 1 \to \emptyset$$

e a árvore tem as arestas

$\{1,2\},\{2,6\},\{6,8\},\{8,3\},\{3,4\},\{4,10\},\{10,5\},\{5,11\},\{11,9\},\{9,13\},\{3,7\},\{7,12\}$

Estes algoritmos podem ser aplicados à solução doutros problemas como a determinação da distância entre dois vértices, a identificação dos vértices de corte do grafo, etc.

1.1.2 Contagem de árvores geradoras

Dado um grafo simples e conexo G qualquer, como determinar o número $t(G)$ das suas árvores geradoras? Para algumas famílias de grafos, muito em particular a dos grafos completos, como veremos mais adiante, é possível obter uma fórmula dependente apenas do número de vértices do grafo. No caso geral, uma das abordagens a este problema passa por encontrar uma relação de recorrência.

Se a é uma aresta de G, designamos por $G \setminus a$ o grafo que se obtém de G eliminando a; e por G/a o grafo que se obtém contraindo a, ou seja, identificando os dois vértices u e v incidentes em a num único vértice w. Note-se que G/a será em geral um multigrafo, uma vez que se em G um vértice z é adjacente a u e a v, em G/a existem duas arestas paralelas entre z e w. Com estas notações, temos então o seguinte resultado:

Proposição 1.8 : *Se G é um grafo e $a \in A_G$,*

$$t(G) = t(G \setminus a) + t(G/a).$$

Demonstração 1.9 : *Se T é uma árvore geradora de G que contém a aresta a, então T/a é uma árvore geradora de G/a e vice-versa.*
Por outro lado, se T é uma árvore geradora de G que não contém a, é também árvore geradora de $G \setminus a$ e vive-versa.

Esta relação pode ser aplicada repetidamente até chegarmos a uma expressão de $t(G)$ como soma de parcelas fáceis de calcular directamente. Algumas observações:

Se a for uma aresta de corte de G (isto é, se $G \setminus a$ for desconexo) a primeira parcela da soma é zero, o que corresponde ao facto de que qualquer árvore geradora de um grafo ter que conter todas as suas arestas de corte.

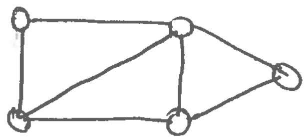
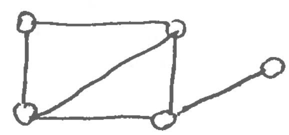
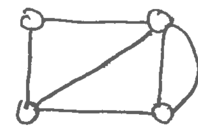
A relação de recorrência pode ser aplicada em qualquer dos sentidos, ou seja, quer considerando uma aresta a de G para eliminar/contrair, quer considerando uma aresta a que *não* está presente em G e acrescentando-a. No primeiro caso usamos a relação na forma $t(G) = t(G \setminus a) + t(G/a)$, e no segundo na forma $t(G \setminus a) = t(G) - t(G/a)$.

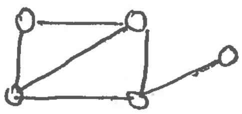
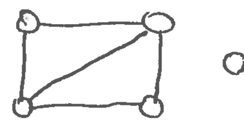
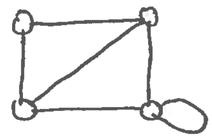
De acordo com o raciocínio feito na demonstração, não podemos ignorar as arestas paralelas que surgem no processo de eliminação/contracção de arestas, porque a escolha de cada uma dessas arestas paralelas para uma árvore geradora corresponde a diferentes árvores geradoras do grafo original; por outro lado, um lacete pode ser ignorado porque nunca faz parte de uma árvore geradora.

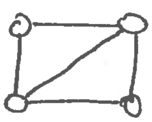
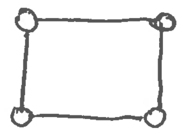
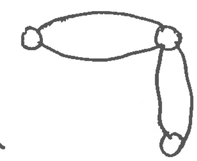
Note-se ainda que se em alguma fase queremos eliminar/contrair arestas paralelas a, a' podemos fazê-lo simultaneamente, adaptando a fórmula: $t(G) = t(G \setminus \{a, a'\}) + 2t(G/\{a, a'\})$.

O exemplo seguinte ilustra a aplicação da fórmula, incluindo de algumas das observações anteriores:

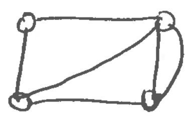
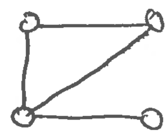
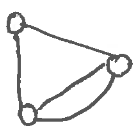
EXEMPLO DE CÁLCULO DO NÚMERO DE ÁRVORES GER

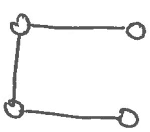
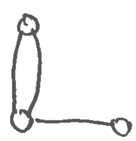
$t\Big($  $\Big) = t\Big($  $\Big) + t\Big($ 

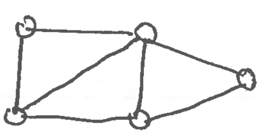
$t\Big($  $\Big) = t\Big($  $\Big) + t\Big($  $\Big)$

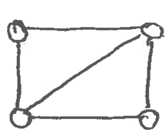
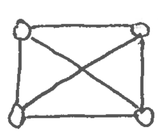
$= 0 + t\Big($  $\Big) \overset{(*)}{=} t\Big($  $\Big) + t\Big($  $\Big)$

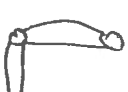
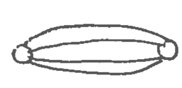
$= 4 + 4 = 8$

$t\Big($  $\Big) = t\Big($  $\Big) + 2 \times t\Big($  $\Big) =$

$= t\Big($  $\Big) + t\Big($  $\Big) + 2 \times 5 = 1 + 2 + 1$

Logo, $t\Big($  $\Big) = 21$

(*) ALTERNATIVA: $t\Big($  $\Big) = t\Big($  $\Big) - t\Big($ 

$= 4^2 - \Big(t\Big($  $\Big) + t\Big($  $\Big)\Big) = 4^2 - (4 +$

1.2 Fórmula de Cayley e Código de Prüfer

Um outro problema de enumeração de árvores é o de calcular o número de árvores com vértices $v_1, v_2, \cdots, v_n$, ou seja, o número $t(K_n)$ de árvores geradoras do grafo completo K_n. A resposta é dada pela

Proposição 1.10 (Fórmula de Cayley)

$$t(K_n) = n^{n-2}$$

Uma das demonstrações desta fórmula foi descoberta por Prüfer e passa por atribuir a cada árvore um código (o código de Prüfer) $a_1 \cdots a_{n-2}$ onde $1 \leq a_i \leq n$.

A definição do código de Prüfer faz-se do seguinte modo: em cada passo, escolhemos a folha (vértice de grau 1) com menor índice, acrescentamos ao código o índice do vértice adjacente e eliminamos daárvore aquela folha e a respectiva aresta incidente; este procedimento é repetido até que tenham sido eliminados $n-2$ vértices e reste portanto uma árvore com dois vértices e uma aresta.
Note-se que o índice i ocorre no código $d(v_i) - 1$ vezes.

Exemplo 1.11 *Se T é a árvore com vértices $v_1, \cdots, v_{13}$ cuja matriz de ad-*

jacência é

$$\begin{pmatrix} 0&0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&0\\ 1&1&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&1&1&0&1&0&1&0&0&0&1&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&1&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&1&0\\ 0&0&0&0&1&0&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix}$$

v_1 é a primeira folha e portanto a primeira entrada do código é 3; a segunda é v_2 e a segunda entrada do código é de novo 3; tendo sido eliminadas as folhas anteriores, v_3 é agora uma folha o que implica que a próxima entrada do código é 5, e assim por diante. Conclui-se que o código de Prüfer de T é

$$[3, 3, 5, 5, 5, 8, 8, 5, 12, 11, 12].$$

Esta definição estabelece uma bijecção entre as árvores de vértices $v_1, \cdots, v_n$ e o conjunto

$$\{[c_1, \cdots, c_{n-2}] : 1 \leq c_i \leq n\}$$

que tem obviamente n^{n-2} elementos.
Para ver que assim é, vamos determinar directamente a aplicação inversa, que a cada sequência faz corresponder uma árvore que tem exactamente essa sequência como código: dada a sequência

$$[c_1, \cdots, c_{n-2}],$$

iniciamos a construção da árvore com duas listas vazias V (vértices) e E (arestas); em cada passo identificamos o primeiro vértice $v \notin V$ e cujo índice

não está na sequência; juntamos v a V, (v, v_c) a E, onde c é a primeira entrada da sequência, e apagamos c da sequência.
Quando a sequência está toda apagada restam dois vértices por escolher que são unidos por uma aresta.

Exemplo 1.12 *: seja a sequência*

$$[2, 5, 1, 1, 10, 7, 2, 3, 3, 5, 3];$$

vamos indicar em cada linha da tabela seguinte a lista V, a aresta criada e o estado da sequência depois de cada passo:

$\{4\}$	$\{v_4, v_2\}$	$[5, 1, 1, 10, 7, 2, 3, 3, 5, 3]$
$\{4, 6\}$	$\{v_6, v_5\}$	$[1, 1, 10, 7, 2, 3, 3, 5, 3]$
$\{4, 6, 8\}$	$\{v_8, v_1\}$	$[1, 10, 7, 2, 3, 3, 5, 3]$
$\{4, 6, 8, 9\}$	$\{v_9, v_1\}$	$[10, 7, 2, 3, 3, 5, 3]$
$\{4, 6, 8, 9, 1\}$	$\{v_1, v_{10}\}$	$[7, 2, 3, 3, 5, 3]$
$\{4, 6, 8, 9, 1, 10\}$	$\{v_{10}, v_7\}$	$[2, 3, 3, 5, 3]$
$\{4, 6, 8, 9, 1, 10, 7\}$	$\{v_7, v_2\}$	$[3, 3, 5, 3]$
$\{4, 6, 8, 9, 1, 10, 7, 2\}$	$\{v_2, v_3\}$	$[3, 5, 3]$
$\{4, 6, 8, 9, 1, 10, 7, 2, 11\}$	$\{v_{11}, v_3\}$	$[5, 3]$
$\{4, 6, 8, 9, 1, 10, 7, 2, 11, 12\}$	$\{v_{12}, v_5\}$	$[3]$
$\{4, 6, 8, 9, 1, 10, 7, 2, 11, 12, 5\}$	$\{v_5, v_3\}$	$[]$

Neste exemplo o processo termina com a criação da aresta $\{v_3, v_{13}\}$.

A verificação de que esta construção é de facto a inversa da do código de Prüfer é deixada para os exercícios.

Uma segunda demonstração da Fórmula de Cayley usa o conceito de **ramificação**.

Definição 1.13 *: Uma ramificação é uma árvore com as arestas orientadas de tal modo que cada vértice é o vértice final de no máximo uma aresta.*

É fácil verificar que, de facto, existe um único vértice (a raiz da ramificação) que não é vértice final de qualquer aresta.

Dado um conjunto de vértices $v_1, \cdots, v_n$, contemos o número de modos de criar uma ramificação, escolhendo uma sequência de $n-1$ arestas: em cada passo, diminuímos o número de componentes conexas numa unidade, e cada uma dessas componentes conexas tem que satisfazer a propriedade da ramificação; no passo k, temos então $n-k+1$ componentes e podemos escolher qualquer um dos n vértices para vértice inicial da aresta; mas o vértice final tem que ser a raiz de uma das outras $n-k$ componentes.
Vemos portanto que há

$$\prod_{k=1}^{n-1} n(n-k) = n^{n-1}(n-1)!$$

construções dessa forma. Mas cada ramificação é construída $(n-1)!$ vezes, uma por cada ordem de escolha das arestas na construção.
Concluímos portanto que existem n^{n-1} ramificações com vértices $v_1, \cdots, v_n$. Mas, para cada árvore nesse conjunto de vértices, existem n ramificações, uma por cada escolha do vértice raiz.

1.3 Grafos com pesos e árvores geradoras minimais

Em certas aplicações, é natural considerar grafos em que a cada aresta é atribuído um determinado valor real (usualmente positivo), um peso. Um problema relacionado com o anterior é o de determinar num grafo conexo com pesos, uma árvore geradora minimal, isto é, com peso total mínimo. Dois algoritmos que resolvem este problema, o de Boruvka-Kruskal e o de Jarník-Prim, baseiam-se no mesmo princípio "ganancioso" de escolher sucessivamente arestas com o menor peso possível.

No algoritmo de Boruvka-Kruskal, a partir de um vértice inicial arbitrário, escolhe-se em cada passo uma aresta de peso o menor possível que não forme um ciclo com outras arestas já incluídas; o algoritmo produz assim uma sucessão de **florestas** que termina com uma árvore geradora minimal.

No algoritmo de Jarník-Prim, também partindo de um vértice inicial arbitrário, escolhe-se em cada passo uma aresta com o menor peso possível, que seja incidente em um vértice já escolhido e noutro ainda não escolhido, ou seja, o algoritmo produz uma sucessão de **árvores**, que termina igualmente com uma árvore geradora minimal.

Para provar que o algoritmo de Boruvka-Kruskal produz uma árvore geradora minimal T, considere-se a lista $a_1, \cdots, a_{n-1}$ das suas arestas, ordenada pela ordem pela qual foram escolhidas, e uma árvore geradora minimal T'. Suponhamos que T' contém o bloco inicial $a_1, \cdots, a_{k-1}$ daquela lista de arestas mas não contém a aresta a_k.
Seja C o (único) caminho em T' que une os vértices incidentes a a_k; C pode conter algumas das arestas $a_1, \cdots, a_{k-1}$, mas tem que conter também arestas fora desse conjunto, caso contrário a_k faria um ciclo com arestas já escolhidas em passos anteriores do algoritmo; vamos ver que, de facto, todas as arestas de C que não estão no conjunto $\{a_1, \cdots, a_{k-1}\}$ têm peso igual ao de a_k:
se existisse alguma aresta a em C com peso menor que $p(a_k)$, ela teria sido escolhida em vez de a_k; note-se de facto que, como quer as arestas $a_1, \cdots, a_{k-1}$ quer a estão em T', a não pode criar um ciclo com aquelas.
Mas, por outro lado, se existisse alguma aresta a em C com peso maior do que $p(a_k)$, poderíamos substituir em T' a aresta a por a_k, obtendo uma árvore com peso menor.
Mas então esta substituição em T' da aresta a por a_k permite obter uma árvore geradora minimal contendo um bloco inicial maior da lista das arestas de T.
Repetindo este procedimento, acabamos por concluir que a própria árvore T é minimal.
Em alternativa, podíamos supor que T' era uma árvore geradora minimal contendo um bloco inicial $a_1, \cdots, a_{k-1}$ das arestas de T o maior possível e, através do raciocínio anterior, chegar a uma contradição.

Provamos que o algoritmo de Jarník-Prim cria umaárvore geradora minimal mostrando, por indução, que a árvore T_m criada após a escolha de m arestas está contida em alguma árvore geradora minimal do grafo.
Fixemos o vértice inicial r; seja $m = 1$ e a_1 a aresta escolhida em primeiro lugar, que liga r ao vértice v_1; se T é uma árvore geradora qualquer que não contém a aresta a_1, existe um caminho (único) em T com início em r e término em v_1; a primeira aresta, incidente em r, desse caminho tem peso igual ou maior ao de a_1; portanto, eliminando essa aresta e acrescentando a_1 obtemos uma nova árvore geradora que tem peso total menor ou igual ao de T; como esta é minimal, o peso tem mesmo que ser igual e temos uma árvore geradora minimal contendo a_1.
Suponhamos agora que a árvore T_{m-1} criada pelo algoritmo após a escolha de $m-1$ arestas está contida nalguma árvore geradora minimal T' e seja a_m a aresta seguinte escolhida pelo algoritmo, ligando algum vértice de T_{m-1} a um novo vértice v_m. Mais uma vez, dado um vértice qualquer de T_{m-1}, existe um único caminho na árvore T' que começa nesse vértice e termina em v_m; seja a a primeira aresta nesse caminho que não pertence a T_{m-1}; essa aresta tem peso maior ou igula ao de a_m (caso contrário, e sendo incidente num vértice já alcançado, teria sido escolhida antes). Substituindo em T' a aresta a pela a_m obtemos, pelo memso argumento do parágrafo anterior, uma árvore geradora minimal que contém a árvore T_m.

Nota 1.14 *Pela sua relação com o tema desta secção, referimos aqui o conhecido "problema do Caixeiro-Viajante" que pode ser descrito como o problema de determinar, dado um grafo completo com pesos nas arestas, um ciclo Hamiltoniano (ou seja, um ciclo que passa por todos os vértices) de peso mínimo. Note-se que este problema contém, como caso particular, o de determinar se um grafo conexo qualquer tem um ciclo Hamiltoniano.*
Apesar da semelhança aparente com o problema de determinar uma árvore geradora minimal, este problema é muito mais complexo. Trata-se, de facto, de um dos vários problemas básicos da Teoria dos Grafos que são NP-completos.

Exercícios X.2

1. Determinar a árvore com vértices $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ cujo código, pelo

algoritmo de Prüfer, é o seu número de aluno, acrescentado de 3 algarismos à escolha.

2. Dados $1 < k < n$ fixos, calcular o número de árvores com vértices $\{v_1, \cdots, v_n\}$ tais que:

   a) $\{v_1, \cdots, v_k\}$ está contido no conjunto das folhas;

   b) $\{v_1, \cdots, v_k\}$ é o conjunto das folhas.

3. Seja a uma aresta de K_n. Mostrar que $K_n - a$ tem $(n-2)n^{n-3}$ árvores geradoras.

4. Mostrar que no código de Prüfer o índice do vértice v ocorre $d(v) - 1$ vezes.

5. Mostrar que se $[c_1, c_2, \cdots, c_{n-2}]$ é o código de Prüfer da árvore T, e v é a folha com menor índice, $[c_2, \cdots, c_{n-2}]$ é o código de Prüfer da árvore $T - v$.

6. Mostrar que os vértices e arestas são eliminados no processo de construção do código de Prüfer pela mesma ordem em que são acrescentados às listas V e E na operação inversa de construção da árvore a partir do código.

7. Dado um grafo conexo G e um vértice v_0, mostrar que existe uma árvore geradora T, tal que, para todo o vértice v, a distância entre v_0 e v em T é igual à distância entre v_0 e v em G.

8. Dado um grafo G conexo com mais do que 2 vértices, justificar que existe um par $x, y \in V_G$ tal que $G - \{x, y\}$ ainda é conexo.

9. Mostrar que o número de árvores com vértices $v_1, \cdots, v_n$ para as quais $d(v_i) = d_i$ é dado por

$$\frac{(n-2)!}{(d_1 - 1)! \cdots (d_n - 1)!}$$

   Sugestão: usar indução em n; notar que podemos assumir que $d_n = 1$. Usando o teorema multinomial, deduzir a fórmula de Cayley.

10. Determinar uma árvore geradora minimal do grafo definido pela seguinte matriz de adjacência com pesos (a entrada A_{ij} é 0 se os vértices v_i ev_j não são adjacentes; caso contrário $A_{ij} = p$ em que p é o peso da aresta que liga os dois vértices):

$$\begin{pmatrix} 0 & 9 & 5 & 8 & 3 & 8 & 7 & 1 \\ 9 & 0 & 8 & 0 & 7 & 4 & 7 & 3 \\ 5 & 8 & 0 & 1 & 0 & 0 & 4 & 8 \\ 8 & 0 & 1 & 0 & 4 & 9 & 3 & 9 \\ 3 & 7 & 0 & 4 & 0 & 3 & 8 & 7 \\ 8 & 4 & 0 & 9 & 3 & 0 & 9 & 6 \\ 7 & 7 & 4 & 3 & 8 & 9 & 0 & 7 \\ 1 & 3 & 8 & 9 & 7 & 6 & 7 & 0 \end{pmatrix}$$

11. Mostrar que se os pesos das arestas de um grafo são todos distintos a árvore geradora com peso mínimo é única.

12. Seja $k > 0$ e T uma árvore qualquer com $k+1$ vértices, dos quais escolhemos um para raiz. Mostrar que se G é um grafo simples com grau mínimo k, para qualquer vértice v de G, existe um subgrafo de G isomorfo a T em que a raiz é v.
    Sugestão: indução.

13. Dados dois conjuntos disjuntos

$$V = \{v_1, \cdots, v_m\}, \qquad U = \{u_1, \cdots, u_n\}$$

calcular o número de árvores com vértices $V \cup U$, em que cada aresta incide num vértice de V e noutro de U, contando as ramificações em $V \cup U$ com raiz em V.

2 Grafos planares

Definição 2.1 : *Um grafo G é* ***planar*** *se admite uma representação gráfica no plano $\mathbb{R}^2$ (designada por representação plana de G) na qual linhas que* *representam arestas diferentes não se intersectam - a não ser nos extremos, no caso das arestas serem incidentes num ou dois vértices comuns.*

Dada uma representação plana de G, identificamos as suas arestas e vértices com os segmentos e pontos que os representam; uma representação plana de G divide o plano em regiões (uma delas ilimitada) que designamos por faces. Essas regiões definem-se rigorosamente como sendo as componentes conexas (por arcos) de $\mathbb{R}^2 \setminus \{\text{representação plana deG}\}$ (ver comentário mais adiante). Por outro lado, a fronteira de cada face de uma representação plana de de G é um passeio fechado do grafo e cada ciclo do grafo resulta, na representação plana, na fronteira de uma união de faces.

Nota 2.2 *Um grafo é planar se e só se cada uma das suas componentes conexas o for.*
Um grafo é planar se e só se o subgrafo simples que resulta de eliminar todos os lacetes e identificar os conjuntos de arestas paralelas também o for.
Um grafo H é uma ***subdivisão*** *de G se for obtido deste introduzindo vértices em arestas de G, isto é, se u e v são adjacentes em G, criamos um novo vértice w e substituímos a aresta $u - v$ pelas arestas $u - w$ e $w - v$. Nestas condições G é planar se e só se H também o for.*
Dado um grafo G, seja G^o o subgrafo que se obtém eliminando sucessivamente os vértices de grau 1 e as arestas neles incidentes até não existirem vértices de grau 1. G é planar se e só se G^o o for.

Um grafo planar tem evidentemente muitas representações planas diferentes. Em particular, podemos escolher uma face qualquer numa representação planar e construir outra representação na qual essa é a face ilimitada. No entanto, o número de faces das representações planas de G é invariante:

Teorema 2.3 ***Fórmula de Euler****: Se G é um grafo planar conexo com v vértices e a arestas e f é o número de faces numa sua representação plana, tem-se*

$$v - a + f = 2$$

Demonstração 2.4 : *Note-se antes do mais que a fórmula é válida para árvores (que são sempre planares): a representação plana de uma árvore tem uma única face e, neste caso,* $v - a = 1$.
Para o caso geral, a demonstração faz-se por indução no número a *de arestas, sendo o caso* $a = 1$ *trivial de verificar. Suponha-se que a fórmula é válida para grafos planares conexos com* m *arestas e seja* G *um grafo planar conexo com* $m+1$ *arestas; se* G *for uma árvore não há nada a demonstrar; caso contrário, dada uma sua representação plana, elimine-se uma aresta que pertença a um ciclo; o grafo* G' *representado continua a ser conexo e tem menos uma aresta que* G *mas a sua representação plana tem também menos uma face que a de* G. *Usando a hipótese de indução aplicada a* G', *concluímos que a fórmula vale também para* G.

2.1 Sólidos regulares

Esta fórmula vale também para os números de faces, arestas e vértices de um sólido poligonal convexo, e permite esclarecer porque existem exactamente 5 sólidos convexos regulares (os chamados sólidos platónicos).
Dado um sólido convexo com f faves, a arestas e v vértices, podemos obter uma representação planar de um grafo cujos vértices, arestas e faces correspondem aos vértices, arestas e faces do sólido. Isso pode fazer-se, por exemplo, por meio da projecção estereográfica: imaginemos o sólido contido no interior de uma esfera, e a projecção das suas arestas e vértices na superfície da esfera, feita a partir do centro da mesma; pousamos a esfera no plano de modo a que o "polo sul" seja o ponto de contacto e que nenhuma aresta (ou vértice) passe pelo "polo norte"; agora projectamos a superfície da esfera no plano, a partir do "polo norte".

Consideremos agora que o sólido é regular e seja p o número de lados e vértices de cada face; se contarmos o número de arestas face a face, como cada aresta é um dos lados de exactamente 2 faces, temos $f \times p = 2a$. Por outro lado, se contarmos as faces vértice a vértice, e se cada vértice pertence a r faces, temos $v \times r = f \times p$. Substituindo na fórmula de Euler

$$2 = v - a + f \implies 2 = \frac{fp}{r} - \frac{fp}{2} + f \implies \frac{p}{r} - \frac{p}{2} + 1 = \frac{2}{f} > 0$$

Dividindo por p obtemos

$$\frac{1}{r} + \frac{1}{p} > \frac{1}{2}$$

cujas soluções são

$r = 3$ e $p = 2$	$f = 4$, $a = 6$ e $v = 4$	(tetraedro);
$r = 3$ e $p = 4$	$f = 6$, $a = 12$ e $v = 8$	(cubo);
$r = 3$ e $p = 5$	$f = 12$, $a = 30$ e $v = 20$	(dodecaedro);
$r = 4$ e $p = 3$	$f = 8$, $a = 12$ e $v = 6$	(octaedro);
$r = 5$ e $p = 3$	$f = 20$, $a = 30$ e $v = 12$	(icosaedro).

2.2 Grafo dual e condições de planaridade

Seja G um grafo planar conexo. Uma vez que cada face de uma representação planar de um grafo corresponde a um passeio fechado do grafo, chamamos grau da face l, e usamos a notação $d(l)$, ao comprimento do passeio correspondente. Se somarmos os graus das faces, cada aresta é contada duas vezes, pelo que concluímos que

$$\sum d(l) = 2a,$$

onde a soma se faz sobre o conjunto das faces de uma qualquer representação planar de G e $a = |E_G|$ é o número das arestas do grafo.

Esta discussão e a fórmula que dela resulta pode ser feita mais precisamente usando a noção de grafo dual: dada uma representação plana de G, define-se o grafo dual G^* do seguinte modo: por cada face da representação de G existe um vértice de G^*; por cada aresta a de G existe uma aresta de G^* que incide nos vértices correspondendo às faces de G que têm a contida na sua fronteira. Note-se que G^* não é em geral um grafo simples: por exemplo, se a for uma aresta contida na fronteira de uma única face (o que acontece se e só se a não

pertence nenhum caminho fechado em G), a aresta correspondente de G^* é um lacete no vértice de G^* respectivo. E se a fronteira comum de duas faces contém várias arestas, os respectivos vértices em G^* são ligados pelo mesmo número de arestas.
Importa notar que o grafo dual depende não apenas de G mas também da representação plana escolhida: duas representações planas do mesmo grafo podem ter duais não isomorfos.

A igualdade $\sum d(l) = 2a$ é portanto apenas o resultado, obtido no início desta introdução à Teoria dos Grafos, que diz que a soma dos graus dos vértices é igual ao dobro do número de arestas, mas aplicado ao grafo dual da representação plana.

Nota 2.5 *Em toda esta discussão, estamos a usar informalmente termos como "lado", "interior" ou "exterior" de uma face. Para formalizarmos estas noções introduzimos algumas definições:*

Definição *2.6* *Uma linha simples em $\mathbb{R}^n$ é a imagem de uma função contínua injectiva $f : [0,1] \to \mathbb{R}^n$. Os pontos $f(0)$ e $f(1)$ são os extremos da linha.*
Uma linha simples fechada é a imagem de uma função contínua $f : [0,1] \to \mathbb{R}^n$ tal que $f(0) = f(1)$ e a restrição de f a $]0,1[$ é injectiva.

Definição *2.7* *Um conjunto $X \subset \mathbb{R}^n$ é conexo por arcos se para todos os $x, y \in X$ existe uma linha simples com extremos x e y contida em X.*

Dito isto, existe o Teorema seguinte (de demonstração menos simples do que se possa supor...)

Teorema *2.8 (Jordan)* *Qualquer linha simples fechada C em $\mathbb{R}^2$ decompôe $\mathbb{R}^2 \setminus C$ na união disjunta de duas regiões não vazias e conexas por arcos.*

Em bom rigor, o Teorema de Jordan não é inteiramente necessário para considerarmos as representações planas de grafos, uma vez que se pode provar que qualquer grafo simples e planar tem uma representação no plano em que as arestas são segmentos de reta. Portanto as faces limitadas da representação são polígonos e nesse caso é mais fácil formalizar a ideia intuitiva de inteiror e exterior.

Suponhamos agora que G é planar, conexo, simples com v vértices, $a > 1$ arestas e f faces. Designando por F o conjunto das faces, e como cada face tem grau pelo menos 3

$$3f \leq \sum_{l \in F} d(l) = 2a;$$

substituindo na fórmula de Euler, obtemos

$$2 = v - a + f \leq v - a + \frac{2}{3}a$$

ou seja:

Proposição 2.9 *Se G é planar, conexo, simples com $v > 2$ vértices e a arestas, então*

$$a \leq 3v - 6.$$

Corolário 2.10 *: O grafo K_5 não é planar.*

Uma generalização do mesmo argumento permite deduzir

Corolário 2.11 *O grafo $K_{3,3}$ não é planar.*

Demonstração 2.12 *O menor ciclo de $K_{3,3}$ tem comprimento 4. Se este grafo fosse planar, dada uma sua representação plana teríamos $4f \leq 2a$ e portanto, repetindo o cálculo anterior,*

$$a \leq 2v - 4$$

Mas $v = 6$ e $a = 9$, o que conduz a uma contradição.

Estes dois exemplos são de facto fundamentais, como se verá a seguir.

2.3 Teoremas de Kuratowski e Wagner

Se alterarmos um grafo G subdividindo uma aresta com um novo vértice, obtemos um novo grafo que se chama uma subdivisão de G; mais geralmente, designa-se **subdivisão de um grafo** G a um grafo que resulte daquele pela repetição dessa operação.

É evidente que um grafo é planar se e só se qualquer sua subdivisão o for. Conclui-se portanto que se um grafo é planar não pode conter como subgrafo uma subdivisão de K_5 ou de $K_{3,3}$. Um notável teorema devido a Kuratowski, diz-nos que esta condição é também suficiente:

Teorema 2.13 *(Kuratowski): Um grafo é planar se e só se não contém como subgrafo uma subdivisão de* K_5 *ou de* $K_{3,3}$.

Um resultado semelhante baseia-se na noçãde menor de um grafo:

Definição 2.14 *Um* ***menor*** *de um grafo* G *é um grafo que se pode obter de* G *através de uma sucessão de eliminação de vértices, e eliminação ou contracção de arestas.*

Teorema 2.15 (Wagner) *Um grafo é planar se e só se não tiver como menor nem* K_5 *nem* $K_{3,3}$

Prova-se que de facto os Teoremas de Kuratowski e Wagner são equivalentes, uma vez que um grafo contém como subgrafo uma subdivisão de K_5 ou de $K_{3,3}$ se e só se tiver esse grafo como menor.

Exercícios X.3

1. Mostrar que se G é um grafo conexo planar com n vértices e m arestas e em que o menor ciclo tem comprimento k, então

$$m \leq k\frac{n-2}{k-2}$$

   Concluir que $Pet(2)$ (ver exercício IX.1.9) não é planar.

2. Mostrar que o grafo complementar de um grafo simples planar com $n \geq 11$ vértices, não é planar.
3. Justificar que se G é um grafo simples planar conexo e $\delta(G) = 5$, então G tem pelo menos 12 vértices de grau mínimo.
4. Mostrar que se um grafo planar G é isomorfo ao seu dual então $|E_G| = 2(|V_G - 1)$.
5. Mostrar que o grafo $Pet(2)$ tem K_5 como menor.