

## Problema E

### Grafos

Arquivo fonte: grafos.{c | cc | java | py2 | py3}

Autor: Prof. Rodrigo Bossini Tavares Moreira

A Teoria dos Grafos consiste no estudo de objetos denominados – quem adivinharia – grafos. É muito fácil definir um grafo matematicamente. Observe: um grafo é um par ordenado  $G=(V, E)$  em que:

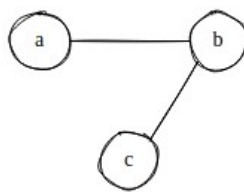
$V$  é seu conjunto de **vértices**

$E \subseteq \{\{x, y\} \mid x, y \in V \wedge x \neq y\}$  é seu conjunto de **arestas**

Viu como é fácil?!

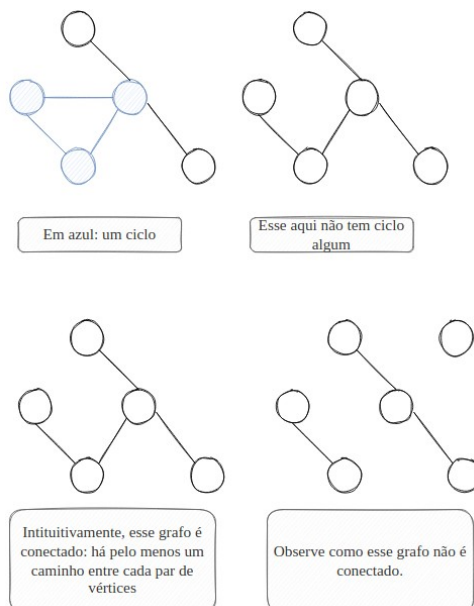
Vejamos um exemplo gráfico. Digamos que  $V=\{a, b, c\}$  e  $E=\{\{a, b\}, \{b, c\}\}$ . A Figura 1 ilustra essa criatura graficamente.

Figura 1



Há diferentes tipos de grafos. Alguns possuem “ciclos” e outros não. Alguns são “conectados” e outros não são. Veja mais alguns exemplos na Figura 2.

Figura 2

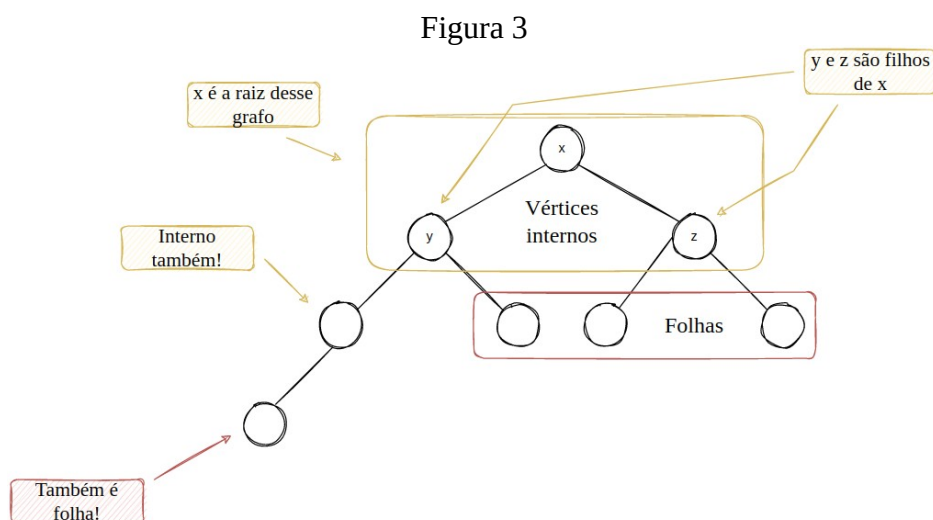


Nem precisa dizer. Você já está achando o assunto super interessante e está ansioso por explorar mais características dos grafos. Nós sabemos. Não é à toa que Leonhard já os estudava nos idos anos de 1700 e alguma coisa.

Entre os diferentes tipos de grafos, há um que se destaca. Ele possui as seguintes propriedades.

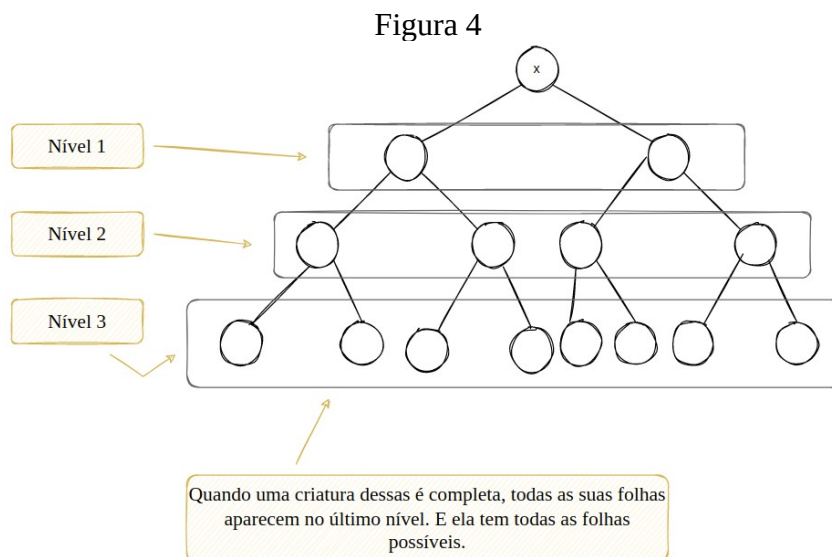
- Um de seus vértices é apontado como especial e denominado raiz.
- É conectado.
- Não possui ciclo algum.
- Cada vértice tem, no máximo, dois filhos. Os filhos de um vértice são vértices também.
- Vértices sem filho algum são chamados folhas.
- Vértices que não são folhas, são vértices internos.

Veja um exemplo de criatura que tem todas essas características na Figura 3.



Raiz...folhas... é grafo mas parece árvore, né? De ponta-cabeça, verdade. Você deve estar se perguntando por que esse tipo de grafo seria tão importante. Ele tem diferentes aplicações: compressão de dados, índices para bancos de dados, ordenação etc. Pode ter certeza de que algum aplicativo ou dispositivo que você usou ou usa no dia a dia emprega uma estrutura assim para resolver algum tipo de problema computacional.

A sua missão neste problema é se aprofundar no estudo sobre essas árvores, digo, grafos, e responder às seguintes perguntas, considerando que eles são **completos**. A Figura 4 mostra um exemplo de grafo (árvore, criatura, whatever) completo(a).



Ao se aprofundar nos estudos, você deverá responder às seguintes perguntas referentes a uma criatura semelhante àquela exibida na Figura 4, considerando que ela tenha **n** vértices.

- Quantos vértices internos ela tem?
- Qual a sua altura?

Ah, antes que nos esqueçamos, a altura de uma criatura assim é o número de **arestas** existentes no caminho entre a raiz e uma de suas folhas. A criatura da Figura 4, por exemplo, tem altura 3.

## Entrada

A primeira linha da entrada contém um número inteiro  $1 \leq t \leq 100$ .  $t$  linhas seguem. Cada linha possui um único número inteiro  $1 \leq n \leq 2^{20}$ , que representa o número de vértices de uma criatura completa, como aquela da Figura 4.

## Saída

Para cada caso de teste, seu programa deve produzir dois números inteiros, numa única linha, separados por um único espaço em branco. O primeiro deve ser o número de vértices internos da criatura. O segundo deve ser a sua altura. Não devem existir quaisquer outros espaços em branco. Também não deve haver linha em branco no final.

## Exemplo

### Entrada

```
2
3
7
```

### Saída

```
1 1
3 2
```