



Número de Erdős

Trabalho Prático

Algoritmos e Programação Orientada a Objetos I

1 Descrição

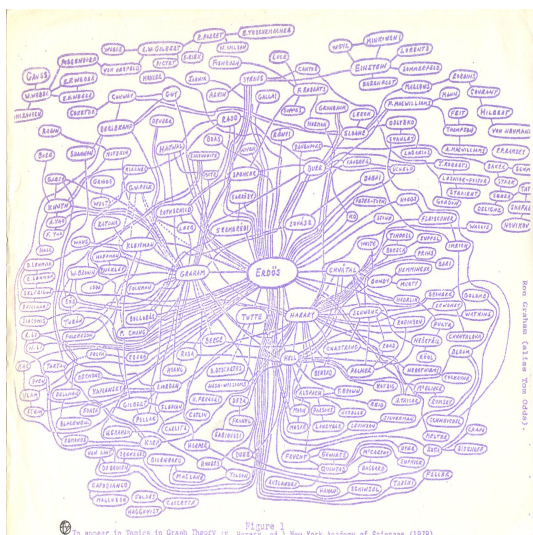


Figura 1: Desenho de Ron Graham.

O matemático húngaro Paul Erdős (*Buda-
peste, Hungria, 1913 – †Varsóvia, Polônia, 1996),
um dos mais brilhantes do século XX, é con-
siderado o mais prolífico matemático da histó-
ria, podendo ser comparado apenas a Leonard
Euler: Erdős publicou mais artigos, enquanto
que Euler publicou mais páginas. Trabalhou
em problemas de combinatória, teoria dos gra-
fos, teoria dos números, análise clássica, teoria
da aproximação, teoria dos conjuntos e teoria
das probabilidades, tendo publicado 1525 arti-
gos em colaboração com 511 outros matemáti-
cos. Em termos de estilo matemático é consi-
derado muito mais um “solucionador de pro-
blemas” do que um “desenvolvedor de teorias”.
Em toda sua carreira, Erdős oferecia prêmios em
dinheiro para soluções de problemas não resol-
vidos. Esses prêmios variavam de \$25, para pro-
blemas mais simples, a vários milhares de dó-

lares para problemas que eram difíceis de atacar e eram matematicamente significativos. O
problema mais conhecido com um prêmio Erdős provavelmente é a conjectura de Collatz, co-
nhecido também como o *Problema $3n + 1$* . Erdős ofereceu 500 dólares por uma solução.

Em homenagem a este gênio húngaro, os matemáticos criaram um número, denominado
número de Erdős. Toda pessoa que escreveu um artigo com Erdős tem o número 1. Todos
que não possuem número 1, mas escreveram algum artigo juntamente com alguém que possui
número 1, possuem número 2. E assim por diante. Quando nenhuma ligação pode ser estabe-
lecida entre Erdős e uma pessoa, diz-se que esta possui número de Erdős infinito. Por exemplo,
o número de Erdős de Albert Einstein é 2. E, talvez surpreendentemente, o número de Erdős
de Bill Gates é 4.

Sua tarefa é escrever um programa que, a partir de uma lista de autores de artigos, deter-
mine o número de Erdős de todos eles.



2 Entrada

Seu programa deve ler vários conjuntos de testes. A primeira linha de um conjunto de teste contém um número inteiro A , com $1 \leq A \leq 100$, que indica o número de artigos. Cada uma das A linhas seguintes contém a lista de autores de um artigo. Cada autor é identificado pela inicial de seu nome, em maiúscula, seguida pelo seu último sobrenome. Por exemplo, P. Erdős. O sobrenome de um autor possui, no máximo, 15 letras, e apenas a letra inicial aparece em maiúscula. Suponha também que não existem letras acentuadas. Assim, consideramos "P. Erdos" ao invés de "P. Erdős". Os autores são separados por vírgulas e a lista de autores de um artigo termina com um ponto. Um único espaço em branco separa a abreviatura do nome do sobrenome, bem como o nome de um autor do anterior. Espaços em branco não são usados em outros locais. Um artigo possui, no máximo, 10 autores, e o total de autores em um caso de teste não excede 100. O final da entrada é indicado por $A = 0$.

3 Exemplo de entrada

```
5
P. Erdos, A. Selberg.
P. Erdos, J. Silva, M. Souza.
M. Souza, A. Selberg, A. Oliveira.
J. Ninguem, M. Ninguem.
P. Duarte, A. Oliveira.

2
Z. Silva, P. Erdos.
Z. Souza.

0
```

4 Saída

Para cada conjunto de teste da entrada seu programa deve produzir um conjunto de linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n ", onde n é numerado sequencialmente a partir de 1. A seguir deve aparecer uma linha para cada autor do conjunto de testes, exceto o próprio P. Erdős. Cada linha deve conter o nome do autor seguido pelo caractere ': ', um espaço em branco e o seu número de Erdős. Caso o número de Erdős de um determinado autor seja infinito, escreva 'infinito'. A saída deve ser ordenada pelo sobrenome do autor e em caso de mesmo sobrenome, o desempate deve ser feito pela inicial do primeiro nome. Imprima uma linha em branco ao final de cada conjunto de teste.



5 Exemplo de saída

```
Teste 1
P. Duarte: 3
J. Ninguem: infinito
M. Ninguem: infinito
A. Oliveira: 2
A. Selberg: 1
J. Silva: 1
M. Souza: 1

Teste 2
Z. Silva: 1
Z. Souza: infinito
```

6 Instruções para entrega

1. Calendário de entrega

O seu trabalho deverá conter pelo menos 2 arquivos: um chamado de `Erdos.java` que contém a classe que tratará o problema de se encontrar o número de Erdős. E uma classe `ErdosMain.java`, onde as funcionalidades implementadas deverão ser testadas. Você pode formular novas classes para auxiliar na resolução.

O trabalho será entregue seguindo a seguinte estratégia:

- **(1,0) Checkpoint 1 - 17/06 das 19 as 21:** Nessa entrega você deve produzir um **diagrama UML** para cada classe que você irá implementar no seu trabalho, com exceção da classe `ErdosMain.java`. Você pode produzir código. Você pode entregar via AVA, no tópico de entrega do trabalho. Você deverá apresentar e explicar para o professor presencialmente suas decisões. As estratégias e possibilidades serão alinhadas nesse dia.
- **(3,0) Checkpoint 2 - 24/06 das 19 as 21:** Nessa entrega você deve produzir os métodos e classes planejados anteriormente. Caso exista alguma modificação, ela deve ser apresentada e justificada. Você pode entregar via AVA, no tópico de entrega do trabalho (no mesmo tópico do anterior!). Erros, dificuldades e possíveis dúvidas deverão ser sanadas. Você deverá apresentar e explicar para o professor presencialmente suas decisões.
- **(6,0) Entrega Final- 26/06:** O trabalho constituído de classes e do diagrama UML produzido.

2. O que entregar?

Seu trabalho também deve ter um cabeçalho, na classe `ErdosMain.java`, com o seguinte formato:



```
/*  
*  
* Nome(s) do(a) (s) estudante(s)  
* RGA(s)  
* Trabalho Prático 1  
* Professor(a): Phelipe Fabres  
*  
*/
```

Os arquivos devem ser compactados em um arquivo zipado com o(s) nome(s) e ultimo(s) nome(s) do(s) aluno(). Por exemplo, se seu nome é **Didier Drogba**, você irá entregar um arquivo zipado com o nome **didier_drogba.zip**.

ATENÇÃO: Você pode usar **apenas o conteúdo aprendido em sala de aula**. No entanto, todos os métodos da classe **String**, mesmo os não vistos, podem ser usados.

ATENÇÃO 2: Você deve usar os métodos de **ordenação** disponibilizados no AVA.

3. Forma de entrega

A entrega do trabalho deve ser realizada diretamente no Sistema de Suporte a Disciplinas da FCOM (AVA), na disciplina de Algoritmos e Programação Orientada a Objetos I - P02. Um fórum de discussão deste trabalho já se encontra aberto. Para entrega do trabalho, você deve estar cadastrado neste ambiente na disciplina Algoritmos e Programação Orientada a Objetos I - P02. Após abrir uma sessão digitando seu *login* e sua senha, vá até o tópico – Trabalho e escolha “Entrega do trabalho”. Você pode entregar o trabalho quantas vezes quiser até às **23 horas e 55 minutos** do dia **26 de Junho de 2019**. A última versão entregue é aquela que será corrigida. Encerrado o prazo, não serão mais aceitos trabalhos.

4. Atrasos

Trabalhos atrasados não serão aceitos. Não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, e/ou falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

5. Erros

Trabalhos com erros de compilação receberão nota **ZERO**. Faça todos os testes necessários para garantir que seu programa está livre de erros de compilação.

6. O que entregar?

Você deve entregar um único arquivo contendo **APENAS** os seus arquivos **.java**. **NÃO** entregue qualquer outro arquivo, tal como o programa executável, já compilado.

7. Verificação dos dados de entrada



Não se preocupe com a verificação dos dados de entrada do seu programa. Seu programa não precisa fazer consistência dos dados de entrada. Isto significa que se, por exemplo, o seu programa pede um número entre 1 e 10 e o usuário digita um número negativo, uma letra, um cifrão, etc, o seu programa pode fazer qualquer coisa, como travar o computador ou encerrar a sua execução abruptamente com respostas erradas.

8. Saída

Observe que a saída do seu programa é enxuta. Em particular, sempre que ocorre alguma interação do programa com o(a) usuário(a), nenhuma mensagem é emitida na saída. Por exemplo, na leitura de um número, não imprima qualquer mensagem como **"Informe um número: "**.

9. Conduta Ética

O trabalho deve ser feito **INDIVIDUALMENTE** ou em **DUPLAS**. Cada estudante tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. Não faça o trabalho em grupo e não compartilhe seu programa ou trechos de seu programa. Você pode consultar seus colegas para esclarecer dúvidas e discutir ideias sobre o trabalho, ao vivo ou no fórum de discussão da disciplina, mas **NÃO** copie o programa!

Trabalhos considerados plagiados terão nota **ZERO**. Estudante que se envolver em **DOIS CASOS DE PLÁGIO** estará automaticamente **REPROVADO** na disciplina.