

# MC558 - Tarefa 2

## 1 Problema

### Código de Prüfer

Dr. Sabin Ada estava trabalhando em um projeto super-secreto para o governo. Neste projeto, ele representava certos objetos através de árvores (não posso dizer exatamente o que são pois o projeto é super-secreto!). Infelizmente, como a maioria dos mortais, sua área do Moogole Drive já estava com taxa de ocupação de 97% (muitas fotos e vídeos de suas façanhas...). Assim, ele não queria armazenar as árvores por listas de adjacências, muito menos por matrizes de adjacências.

Dr. Ada lembrou-se então de suas aulas de Teoria dos Grafos, em particular, do código de Prüfer. O resultado diz que existe uma bijeção  $f$  entre a família das árvores com conjunto de vértices  $S$  ( $|S| = n$ ) e  $S^{n-2}$ , o conjunto das  $(n-2)$ -uplas com elementos em  $S$ . Assim, ele teve a “brilhante” ideia de representar cada árvore  $T$  que ele tinha pelo seu código  $f(T)$ . Ele escreveu um programa que dado um código  $(a_1, \dots, a_{n-2}) \in S^{n-2}$ , este reconstruía a árvore  $T$  tal que  $f(T) = (a_1, \dots, a_{n-2})$ . Com isto, ele sempre poderia recuperar a árvore, caso fosse necessário.

Obviamente, uma semana depois seu espaço disponível acabou e o Moogole Drive decidiu apagar seu programa. Dr. Ada acabou conseguindo mais espaço depois, mas seu programa estava perdido. Sua tarefa é implementar este programa. Para maiores detalhes sobre a função  $f$ , consulte os slides.

## 2 Entrada e Saída

Suporemos que o conjunto de vértices da árvore é  $S = \{1, 2, \dots, n\}$ .

**Entrada:** a entrada é composta de duas linhas. A primeira linha contém um inteiro  $2 \leq n \leq 1000$ . A segunda linha contém  $n-2$  inteiros em  $S$  que correspondem  $(n-2)$ -upla  $a = (a_1, \dots, a_{n-2})$ . Os elementos estão separados por um espaço.

**Saída:** a saída é a árvore representada pelo **vetor de predecessores** (visto em aula). Assim, o  $i$ -ésimo elemento impresso deve ser o predecessor do vértice  $i$ . Os elementos devem estar em uma única linha e devem ser separados por um espaço (há uma quebra de linha no fim). A **raiz** da árvore deve ser o **menor** dos dois vértices que sobram no final: a última aresta conecta estes dois vértices. Considere que o predecessor da raiz tem rótulo 0.

**Exemplos:**

<b>Entrada</b>	<b>Saída</b>
5 1 1 2	2 0 1 1 2
8 7 4 4 1 7 1	0 7 4 1 4 7 1 1

### **3 Implementação e Submissão**

- A solução deverá ser implementada em C, C++, Pascal ou Python 2/Python 3. Só é permitido o uso de bibliotecas padrão.
- O programa deve ser submetido no SuSy, com o nome principal **t2** (por exemplo, t2.c).
- O número máximo de submissões é 20.
- A tarefa contém 10 testes abertos e 10 testes fechados. A nota será proporcional ao número de acertos nos testes fechados.

### **4 Prazo**

A solução pode ser submetida até o dia **03/05/21**.