

Algoritmos e Estruturas de Dados 2

Terceiro Trabalho Prático (Segunda Versão)

Entrega Sugerida: 01/10/2022

1º Semestre 2022 - DC-UFSCar

1 Introdução

No terceiro Trabalho Prático (TP03) será solicitada a entrega de um programa que solucione o problema apresentado na próxima seção. O arquivo entregue deve seguir os itens abaixo:

- O TP03 deverá ser feito individualmente e plágio não será tolerado;
- O TP03 deve ser entregue no run codes (<https://run.codes>) em um arquivo contendo código em linguagem C, e com um cabeçalho com as informações do estudante (nome, curso, RA);
- Cada estudante deve se cadastrar no run codes (<https://run.codes>) informando Nome Completo, escolhendo “UFSCar - Universidade Federal de São Carlos” no campo Universidade e colocando seu RA no campo Núm. Matrícula. Depois de cadastrado, basta logar no run codes e se matricular na disciplina “1001490 - Algoritmos e Estruturas de Dados 2” usando o Código de Matrícula J81A;
- Compile o seu TP usando o compilador GCC com flags -Wall -pedantic -O2 -Wno-unused-result, pois warnings podem impedir o código de funcionar no run codes ainda que funcionem no seu computador.
- Importante1: para todos os estudantes de computação (engenharia ou ciência), gostaria que o trabalho fosse feito em linguagem C, como está especificado no enunciado.
- Importante2: No run codes eu também liberei submissão em Python apenas para que alguns estudantes da estatística, que estão cursando a disciplina, possam usar essa linguagem, já que eles não veem introdução à programação em C.
- Importante3: Os estudantes da estatística (ou de outros cursos que não de computação), por favor me enviem uma mensagem para eu lembrar de fazer o ”if” de linguagem na correção dos trabalhos.

2 Debate

Voltemos à quinta série. Não, não vamos fazer trocadilhos ofensivos. Neste problema você deve ajudar o senhor William, professor de Geografia que adora fazer debates entre os alunos.

Como atividade de suas aulas, William decidiu fazer um debate entre os alunos separando-os em dois grupos, os quais devem defender ideias opostas. Para montar esses grupos, o professor pediu para que cada aluno anotasse o nome de um ou mais estudantes para os quais deseja fazer perguntas. Mas, para isso, há a restrição de que uma pergunta deve ser feita por alguém de um grupo e respondida por algum participante do grupo oposto. Não deve haver interação entre membros do mesmo grupo.

Por exemplo, considere que dois grupos devem ser formados com os alunos João, Paulo, Maria e Sara. Caso João queira perguntar a Paulo e Maria, João deve estar em um grupo, enquanto Paulo e Maria em outro. Assim, Paulo e Maria não poderão fazer perguntas um ao outro. Nesse cenário, Sara não poderá perguntar a João e também a alguém do grupo com Paulo e Maria, senão não poderia ser colocada em qualquer grupo sem ferir as restrições. Caso Sara decida por fazer perguntas apenas a Paulo, por exemplo, ela será do grupo de João.

Sua tarefa é ajudar o professor a decidir se é possível separar os alunos em dois grupos sem ferir qualquer restrição. Para isso, você deve fazer um programa que receba a lista de alunos, identificados por seu número de chamada, e seus interesses em fazer perguntas. Ao final, responda se é possível ou não tal divisão.

Entrada

Os dados de cada caso de teste estão armazenado em um arquivo com extensão “.in”. Assim, a única linha a ser lida da entrada padrão contém o nome de tal arquivo. Esse nome deve ser usado para ler o arquivo, que se encontra na mesma pasta que o executável.

A primeira linha de cada arquivo contém um único inteiro positivo, indicando quantas instâncias estão contidas no arquivo. A primeira linha de cada instância indica o número N ($2 \leq N \leq 1000$) de estudantes na turma a ser analisada. As próximas N linhas contém informações sobre a vontade de cada estudante fazer perguntas. Mais precisamente, a primeira das N linhas corresponde ao estudante 0, a seguinte ao estudante 1, e assim por diante. Cada uma dessas linhas começa com um inteiro M ($1 \leq M \leq N/2$) que indica a quantos alunos o estudante corrente deseja fazer perguntas. Esse inteiro M é seguido, na mesma linha, de M valores correspondentes aos números dos alunos alvos das questões. Note que os números de identificação dos estudantes começam em 0 e vão até $N - 1$.

Saída

Para cada instância do arquivo com os dados do caso de teste, seu programa deve imprimir a palavra “Impossível” em uma nova linha, caso não seja possível

formar os dois grupos sem qualquer restrição. Caso seja possível, seu programa deve imprimir a frase “Vai ter debate” em uma nova linha.

Exemplo de entrada

```
2
10
3 1 3 5
2 6 8
2 8 4
1 2
4 0 2 6 8
2 3 7
1 1
2 6 2
1 1
3 1 2 3
6
2 4 5
1 3
2 3 5
1 0
2 2 1
1 1
```

Saída esperada para o exemplo

```
Impossible
Vai ter debate
```