

Instituto de Computação - Unicamp

MC558 - Projeto e Análise de Algoritmos II

Laboratório 01 - Blueprints

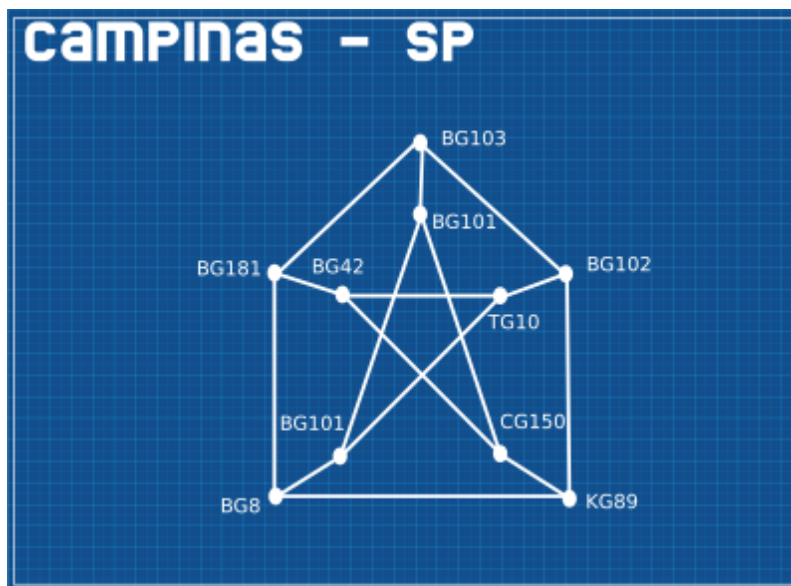
Prazo de entrega: **19/09/2016 às 23:59:59**

Professor: Orlando Lee

Monitores: Guilherme Bueno Andrade (PAD) e Maycon Sambinelli (PED)

Descrição

A SMART Telecom é uma empresa de telefonia fixa que atende diversas cidades brasileiras. Para cada cidade atendida, a SMART Telecom mantém uma *blueprint* da sua rede. Nesta blueprint, podemos ver as centrais de telefonia instaladas na cidade e como elas estão conectadas entre si. Veja abaixo um exemplo.



Essas blueprints são impressas em grandes folhas de A1 e guardadas em um grande armário na sede da empresa. Infelizmente, uma forte chuva quebrou o telhado da sede e encharcou esse armário. Apesar de todos os

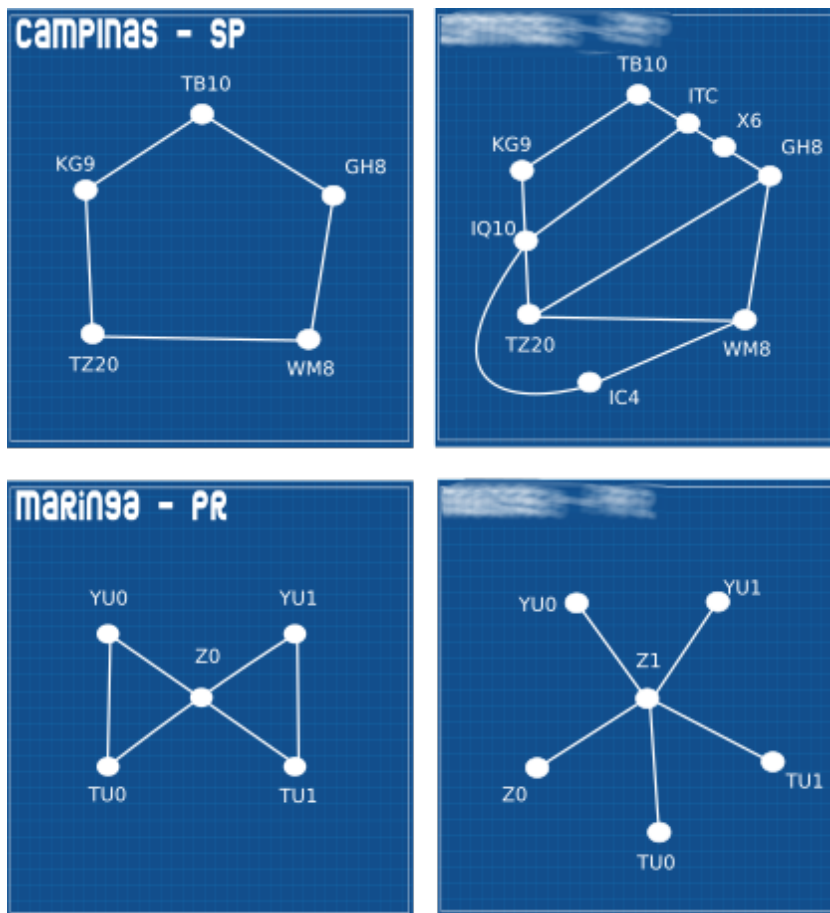
detalhes técnicos terem sido preservados, as identificações das blueprints estão totalmente borradas e por isso não é possível saber a qual cidade pertence cada blueprint.

A SMART Telecom conseguiu encontrar versões antigas e desatualizadas das blueprints em um arquivo morto que não foi afetado pela chuva e pretende basear-se nessas blueprints para identificar as danificadas pela chuva.

Baseando-se em suas políticas de expansão de rede, ela definiu o seguinte critério para decidir se duas blueprints são da mesma cidade: dada uma blueprint A cuja identificação foi danificada pela chuva e uma blueprint B da cidade X que estava no arquivo morto, dizemos que A é uma blueprint da cidade X se:

1. Todas as centrais presentes na instalação B estão presentes em A.
2. Se existir uma conexão entre uma central y e uma central w na blueprint B então deve existir uma rota de conexões (uma sequência x_1, x_2, \dots, x_n de centrais tais que existe uma conexão entre as centrais x_i e x_{i+1} para $i = 1, 2, \dots, n - 1$) da central y a central w na blueprint A tal que qualquer central interna (qualquer central diferente de x_1 e x_n) é uma central nova (não pertence a blueprint B).

A seguir apresentamos dois exemplos, um por linha, de blueprints de uma mesma cidade (onde a blueprint da esquerda é a encontrada no arquivo morto enquanto que a da direita é a danificada pela chuva).



Como essa identificação estava levando muito tempo, a SMART Telecom contratou você para ajudá-los com essa tarefa através do desenvolvimento de um programa que recebe como entrada a descrição de duas blueprints e decide se elas são da mesma cidade ou não.

Entrada

A entrada consiste de:

1. Um número inteiro m_1 ($1 \leq m_1 \leq 50$), que representa o número de conexões da blueprint encontrada no arquivo morto.
2. Seguida de m_1 linhas, onde cada linha contém o código de duas centrais separadas por espaço, indicando que existe uma conexão entre elas na blueprint encontrada no arquivo morto.
3. Seguido de um número inteiro m_2 ($1 \leq m_2 \leq 50$), que representa o número de conexões da blueprint danificada pela chuva.
4. Seguida de m_2 linhas, onde cada linha contém o código de duas centrais separadas por espaço, indicando que existe uma conexão entre elas na blueprint danificada pela chuva.

Você pode assumir que toda blueprint possui no máximo 500 centrais e no máximo 10000 conexões.

Saída

A saída esperada consiste de uma única linha contendo a palavra 'SIM', no caso do das duas blueprints pertencerem a mesma cidade, e 'NAO', caso contrário.

Exemplos

Exemplo 1

- Entrada

```
4
EI232 CJ421
EI232 EI402
EJ203 FG133
EI402 CJ421
7
CK003 CG401
EJ203 CG401
EI402 CK003
CJ421 CG401
EI232 CG401
FG133 CG401
EI232 CK003
```

- Saída

```
SIM
```

Exemplo 2

- Entrada

```
7
AG404 EG120
```

FJ130 DH342
AG404 FG222
DG012 DH342
EG120 DG012
DG012 FJ130
FG222 EG120
9
FJ130 DH342
FG222 EG120
DH342 CI332
DG012 CI332
AG404 FG222
DG012 FJ130
DG012 DH342
AG404 EG120
EG120 EJ334

- Saída

NAO

Para mais exemplos, consulte os [testes abertos no Susy](#).

Restrições

O seu programa deve atender as seguintes restrições:

- Complexidade: $O(m_1 n_2^2 + m_2 n_2)$, onde
 - m_1 é o número de conexões da blueprint encontrada no arquivo morto;
 - n_2 é o número de centrais da blueprint danificada pela chuva; e
 - m_2 é o número de conexões da blueprint danificada pela chuva.

Relatório

Além do programa, você deve escrever um pequeno relatório explicando a ideia utilizada para resolver o problema e fazer a análise de complexidade de tempo do seu programa (não é necessário provar a corretude do algoritmo). O relatório deve conter no máximo uma página, deve estar no formato pdf e deve ser submetido pelo SuSy.

Critérios de avaliação

A nota máxima do laboratório é 10 e é dada pela seguinte formula:

$$NF = NP + NR - PE$$

Onde:

- NF é a nota final.
- NP é a nota do programa. Esse valor é igual a 6 se o programa passou em todos os casos de teste do SuSy. No caso do programa ter falhado em um ou mais casos de teste, temos que $NF = 0$ (Note que neste caso é a nota final que é igual a zero e não a nota do programa).
- NR é a nota do relatório que é dado pela soma da nota dada pela explicação da ideia (2 pontos) mais a nota dada pela análise de complexidade do programa (2 pontos).
- PE é a soma das penalidades aplicadas ao programa. O valor das penalidades é apresentado na tabela abaixo.

Valor	Descrição
-2	Programa com problemas relevantes de qualidade de código (Falta de comentários, nomes não significativos e etc)
-3	Programa que violar alguma das restrições apresentadas na seção <i>Restrições</i>

Observações

- O número máximo de submissões é **15**;
- O seu usuário no SuSy é o seu número de RA (apenas números) e a sua senha é a sua senha da DAC.
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.
- O SuSy utiliza as seguintes flags de compilação: `-std=c99 -pedantic -Wall -lm` para a linguagem C e `-ansi -pedantic -Wall -lm` para a linguagem C++.
- Para efeito de avaliação será levado em conta apenas a última submissão no SuSy. Arquivos fontes mandados por email não serão

levados em conta.

Plágio

O reaproveitamento de código da Web ou de colegas é considerado plágio e será tratado de acordo com os critérios estabelecidos.