



**Universidade de São Paulo**  
Instituto de Ciências Matemáticas e de Computação  
Departamento de Ciências da Computação  
SCC0218 — Algoritmos Avançados e Aplicações

## Trabalho 02: Projeto Cartógrafo Cósmico, Uma Tarefa para Lovelace

**Professora:** Leo Sampaio Ferraz Ribeiro  
**Estagiária PAE:** Raissa Rosa dos Santos Januário  
**Pessoas Monitoras:** Clara Ernesto de Carvalho e Lucas Henrique Sant’Anna

Desenvolva o trabalho sem olhar o de colegas.  
Se precisar de ajuda pergunte, a equipe de apoio está aqui por você.

### Introdução

Olá, tripulante da *Andarilha*! A nossa IA de bordo, Lovelace, precisa da sua ajuda em uma tarefa de programação de alta complexidade. O Capitão Ashby aceitou um novo contrato da Comunidade Galáctica (CG) para mapear um aglomerado estelar recém-descoberto e estabelecer uma nova rede de túneis subespaciais.

Para garantir que a rede seja segura e eficiente, a sua missão é desenvolver um algoritmo em duas fases. Primeiro, você ajudará Lovelace a projetar a rota mais estável que conecte os sistemas mais importantes. Em seguida, identificará o maior risco de navegação no aglomerado. Ohan já nos forneceram os dados dos sistemas estelares; agora, a lógica de programação é por sua conta.

### Descrição do Problema

O mapa estelar é um plano 2D, onde cada sistema é um ponto com coordenadas  $(x, y)$ .

#### Parte 1: A Malha de Túneis Essencial (Algoritmo Guloso)

A primeira diretriz do Capitão Ashby é criar a “Malha de Túneis Essencial”. Esta é uma rede de túneis que deve conectar os sistemas estelares mais importantes do aglomerado com o menor custo total de tensão subespacial. A tensão de um túnel é simplesmente a distância Euclidiana entre os dois sistemas que ele conecta. Uma tensão maior que certo valor implica que um túnel não pode existir entre este par de sistemas.

**Seu Objetivo:** Implementar uma função que receba a lista de sistemas estelares e seleciona as conexões para formar a Malha de Túneis com o menor custo total de tensão subespacial. Você deve resolver este problema usando um algoritmo guloso.

## Parte 2: O Ponto de Ressonância Gravitacional (Algoritmo de Divisão e Conquista)

Com a malha de túneis planejada, Sissix, nossa piloto, levanta uma preocupação de segurança crucial. Sistemas estelares que estão muito próximos um do outro podem criar uma ressonância gravitacional, gerando instabilidade em toda a região e tornando a navegação perigosa. Lovelace precisa identificar o par de sistemas mais próximo em todo o aglomerado para que Sissix e Jenks possam marcá-lo como uma zona de alerta máximo.

**Seu Objetivo:** Implementar uma função que encontre o par de sistemas estelares mais próximo em todo o aglomerado. Você deve resolver este problema usando um algoritmo de divisão e conquista. Note que aqui consideramos todos os sistemas do aglomerado e não apenas os mais importantes, então nada de reutilizar as distâncias calculadas no passo anterior.

### Entrada

Primeira linha com o número de problemas a serem resolvidos. Cada problema então será composto pelo **número de sistemas total**, **número de sistemas mais importantes** e **valor máximo de tensão subespacial** que ainda permite a existência de um túnel; após estas informações segue uma lista de sistemas estelares, começando pelos mais importantes. Cada sistema será representado por um identificador do sistema (uma **string**) e as coordenadas  $(x, y)$  do sistema no espaço. Exemplo:

```
2
5 4 40
Aganon 0 1
Akari 2 3
Arun 40 10
Enceladus 20 5
Tren 0 1.5
3 3 20
Akari 2 3
Enceladus 20 5
Aganon 0 1
```

### Saída

Seu programa deve produzir as saídas para ambas partes, uma após a outra:

- **Para a Parte 1 (Malha de Túneis Essencial):** Uma lista com os túneis selecionados para a malha de túneis, listando os identificadores de ambos sistemas (em

ordem alfabética) e a tensão subespacial entre eles:  
id\_sistema\_1, id\_sistema\_2, tensao

- **Para a Parte 2 (Ponto de Ressonância):** a distância mínima e os identificadores dos dois sistemas mais próximos:

Ponto de Ressonância: id\_sistema\_1, id\_sistema\_2, distancia\_minima

Use uma quebra de linha para separar as diferentes instâncias do problema:

Aganon, Akari, 2.83  
Akari, Enceladus, 18.11  
Arun, Enceladus, 20.62  
Ponto de Ressonância: Aganon, Tren, 2.83

Akari, Aganon, 2.83  
Akari, Enceladus, 18.11  
Ponto de Ressonância: Akari, Aganon, 2.83

## CrITÉRIOS de Avaliação

Faça uso dos paradigmas de programação indicados, implemente um código legível e de fácil interpretação e atinja a complexidade esperada para cada parte:

- **Parte 1:** A complexidade esperada é  $O(E \log E)$  ou  $O(E \log V)$ , onde  $E$  é o número de arestas (túneis) e  $V$  o de vértices (sistemas).
- **Parte 2:** A complexidade esperada é  $O(N \log N)$ , onde  $N$  é o número de sistemas estelares.

## Submissão

1. **Envie** seu código fonte para o run.codes.
2. **Tire Dúvidas com a Equipe de Apoio.** Se não conseguiu chegar em uma solução, dê um tempo para descansar a cabeça e converse com a equipe de apoio sobre a dificuldade encontrada.