Grafos Eurelianos e Hamiltonianos

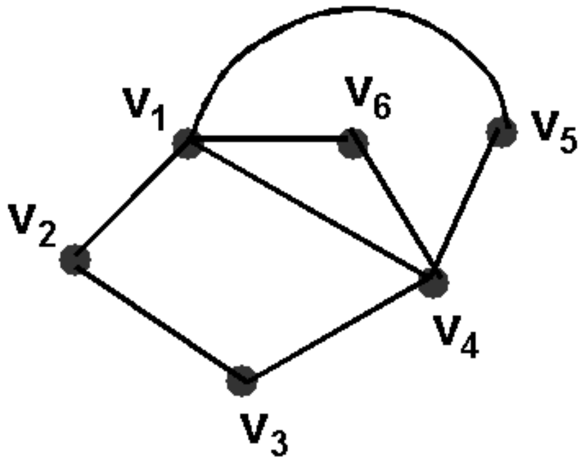
Profa. Dra. Cristina Dutra de Aguiar

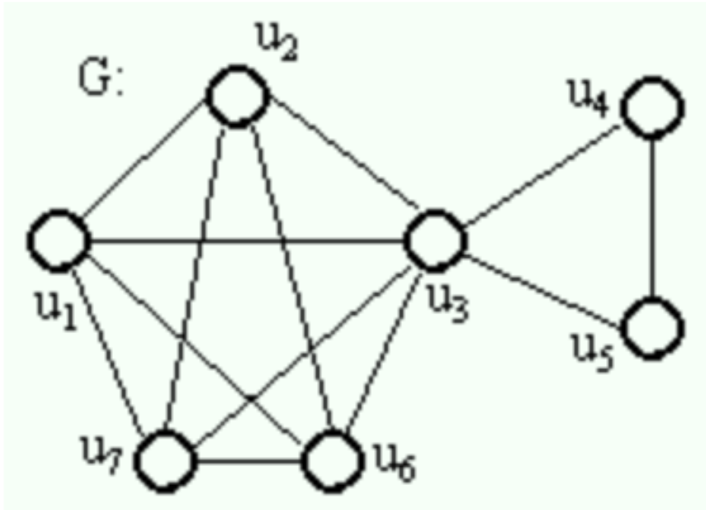
Definições

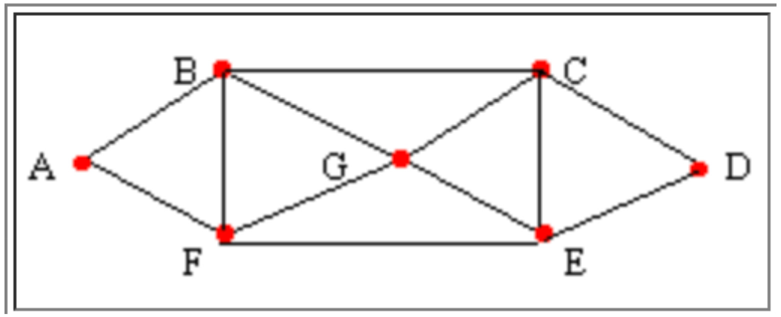
- Grafo euleriano
  - é possível visitar todas as arestas, porém passando em cada aresta apenas uma vez
  - o vértice origem e o vértice destino são os mesmos (forma-se um ciclo eureliano)

Um grafo conexo G é eureliano se e somente se cada vértice de G possui grau par

Exemplos







Definições

- Caminho euleriano
  – caminho que inclui cada aresta do grafo exatamente uma vez
- Ciclo eureliano
  – ciclo que inclui cada aresta do grafo exatamente uma única vez

Definições

- Grafo semi-euleriano
  - é possível visitar todas as arestas passando em cada aresta apenas uma vez
  - o vértice destino é diferente do vértice origem. Neste caso, um é o ponto de partida e o outro é o ponto de chegada

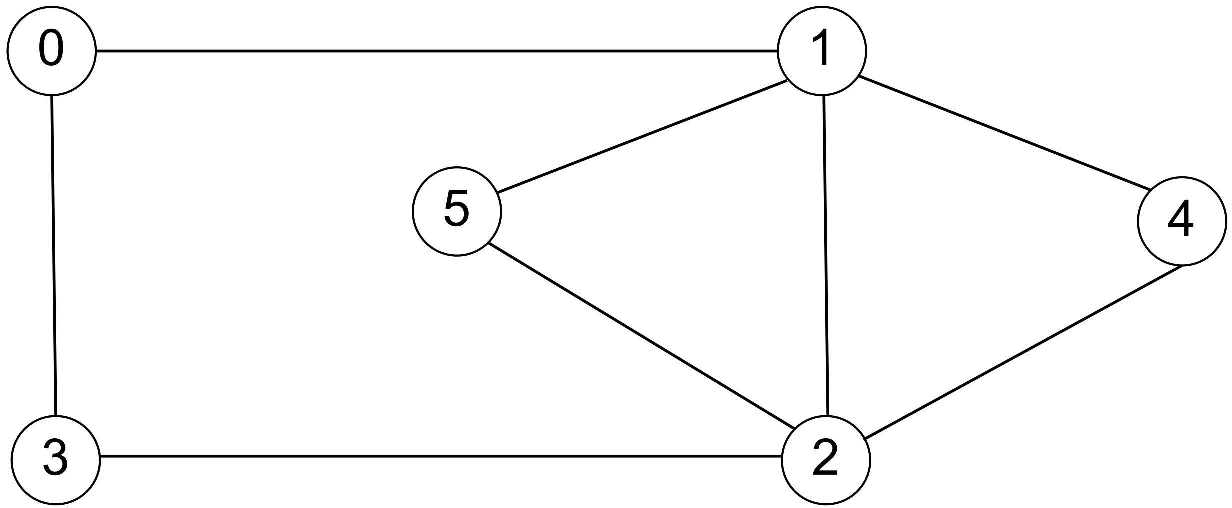
Um grafo conexo G é semi-eureliano se, e somente se, possuir no máximo um par de vértices de grau ímpar

Algoritmo de Hierholzer

Dado um grafo eureliano, determina o ciclo eureliano

```
S: primeira lista de vértices // armazena os vértices em análise
T: segunda lista de vértices // contém o ciclo eureliano
desmarque todas as arestas
escolha um vértice inicial v e adicione-o em S
enquanto S não estiver vazio
   considere u o último vértice de S
   se existe uma aresta (u,v) desmarcada que incide em v
       então adicione v a S e marque a aresta (u,v)
       senão retire u de S e adicione-o em T
fim enquanto
```

Algoritmo de Hierholzer



vértice inicial: 0

ciclo eureliano: T = { 0, 3, 2, 5, 1, 4, 2, 1, 0 }

Problema do Carteiro Chinês

- Elaborado em 1962 pelo matemático chinês Mei-Ku Kwan
- Dado um grafo G = (V, A) ponderado (direcionado ou não)
  - encontre um ciclo eureliano de custo mínimo

Exemplos de Aplicação

- Entrega de correspondências
  - um carteiro deve percorrer as ruas de um bairro entregando as cartas em todas as ruas do bairro, começando e terminando no ponto de distribuição
- Outros exemplos
  - coleta de lixo doméstico
  - nebulização de combate à dengue
  - recenseamento
  - vendas em domicílio

Solução

Se o grafo é eureliano

então aplicar um algoritmo de determinação
de ciclo eureliano (ex. Hierholzer e Fleury)

senão acrescentar arestas artificiais
// transformar o grafo original em um
// supergrafo (vértices de grau ímpar
// devem ser transformados em vértices de
// em vértices de grau par)
aplicar um algoritmo de determinação
de ciclo eureliano (ex. Hierholzer e Fleury)

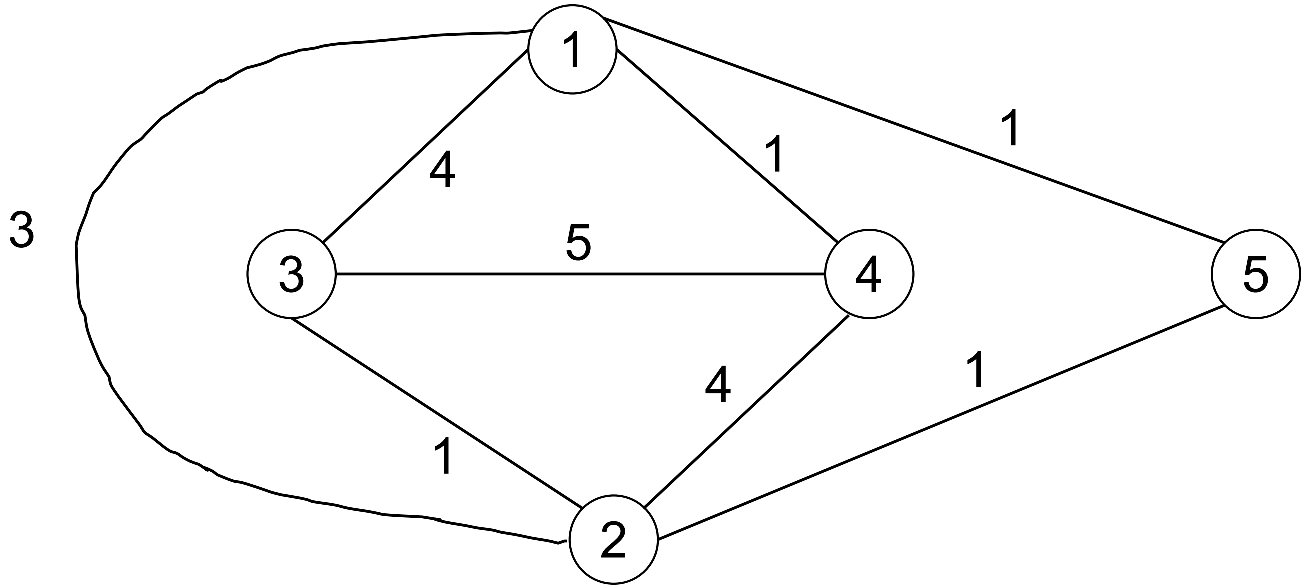
Transformação do Grafo

- Grafo G é transformado no supergrafo G*
  - adiciona arestas paralelas
    - ou seja, ocorre a duplicação de arestas
  - de forma a minimizar o somatório dos pesos das arestas que estão em G* mas não estão em G
    - duplica as arestas, porém minimizando a soma dos pesos das arestas duplicadas

Acrescentar Arestas

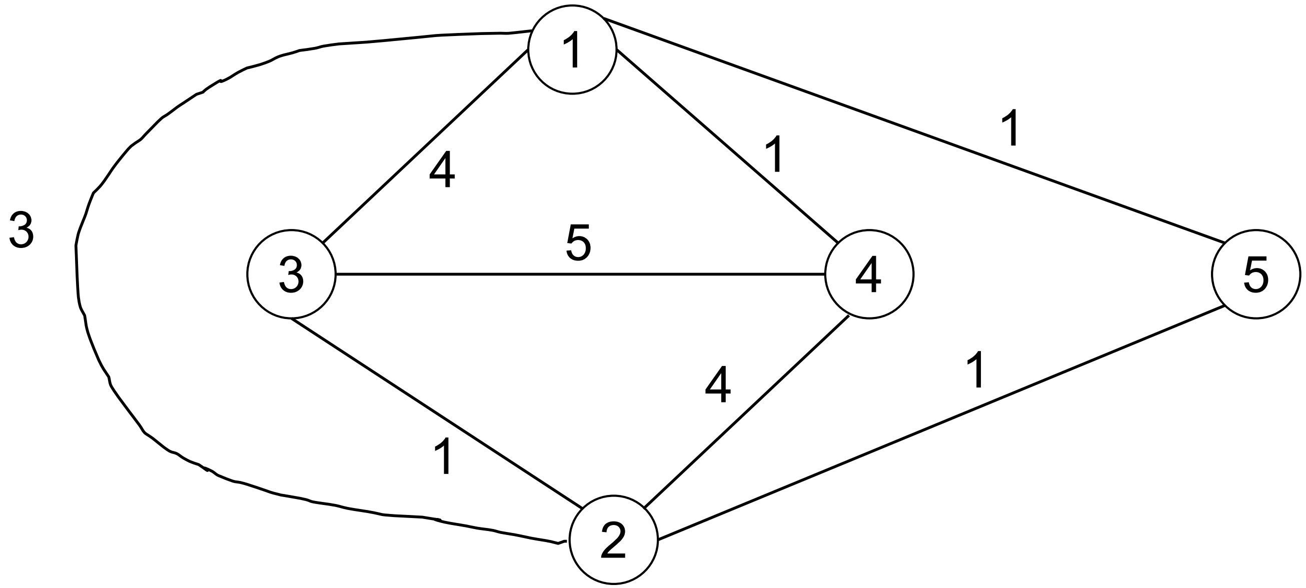
- Grafo com dois vértices de grau ímpar
  - acrescentar aresta com o custo do menor caminho entre os vértices (Dijkstra)
- Grafo com 4 ou mais vértices de grau ímpar
  - montar um grafo completo com esses vértices, sendo que cada aresta representa o menor caminho entre os pares de vértices (Floyd)

Exemplo



O grafo G é um grafo eureliano?

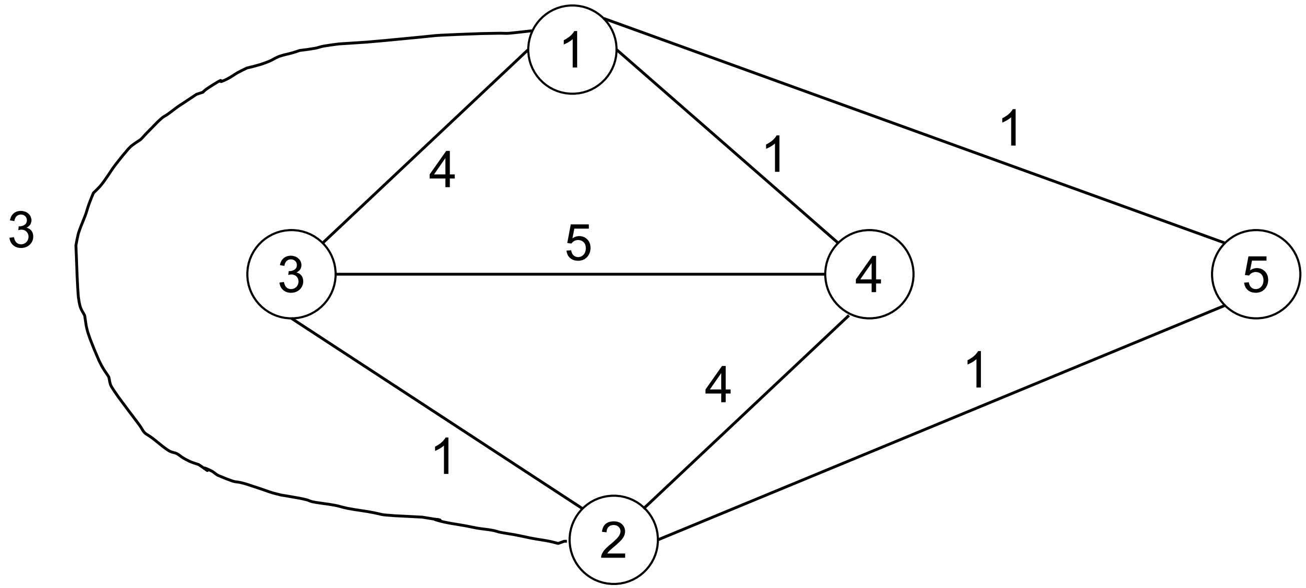
Exemplo



O grafo G é um grafo eureliano? não, desde que os vértices 3 e 4 possuem grau ímpar

Ação: obter o menor caminho entre os vértices 3 e 4 usando Dijkstra

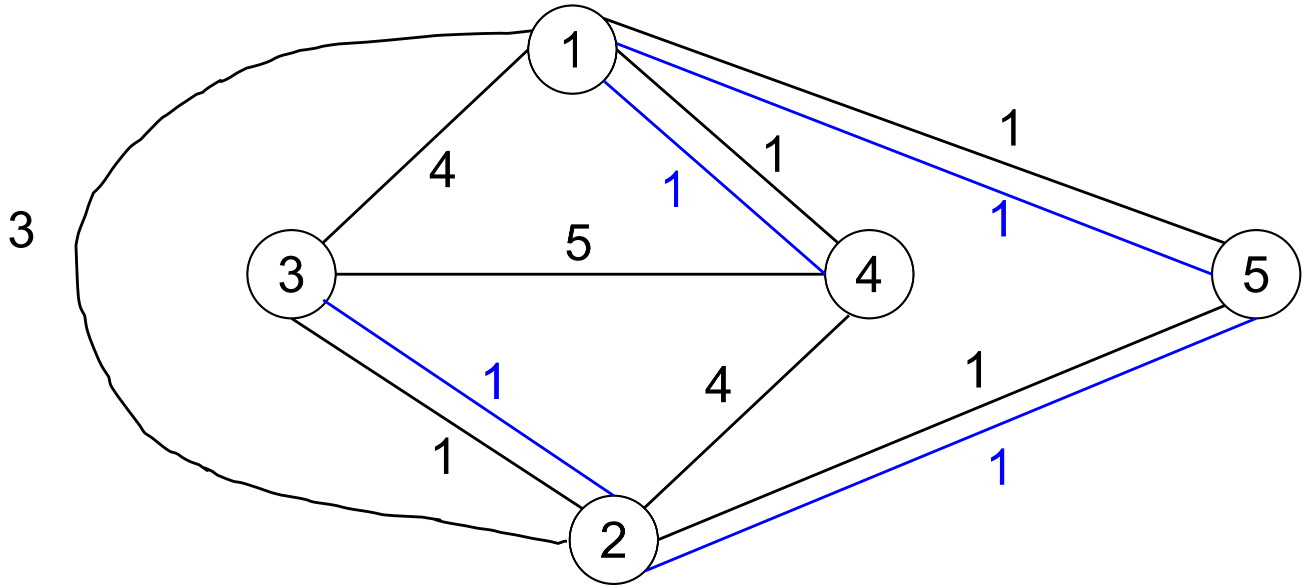
Exemplo



Menor caminho entre os vértices 3 e 4 usando Dijkstra: 3, 2, 5, 1, 4

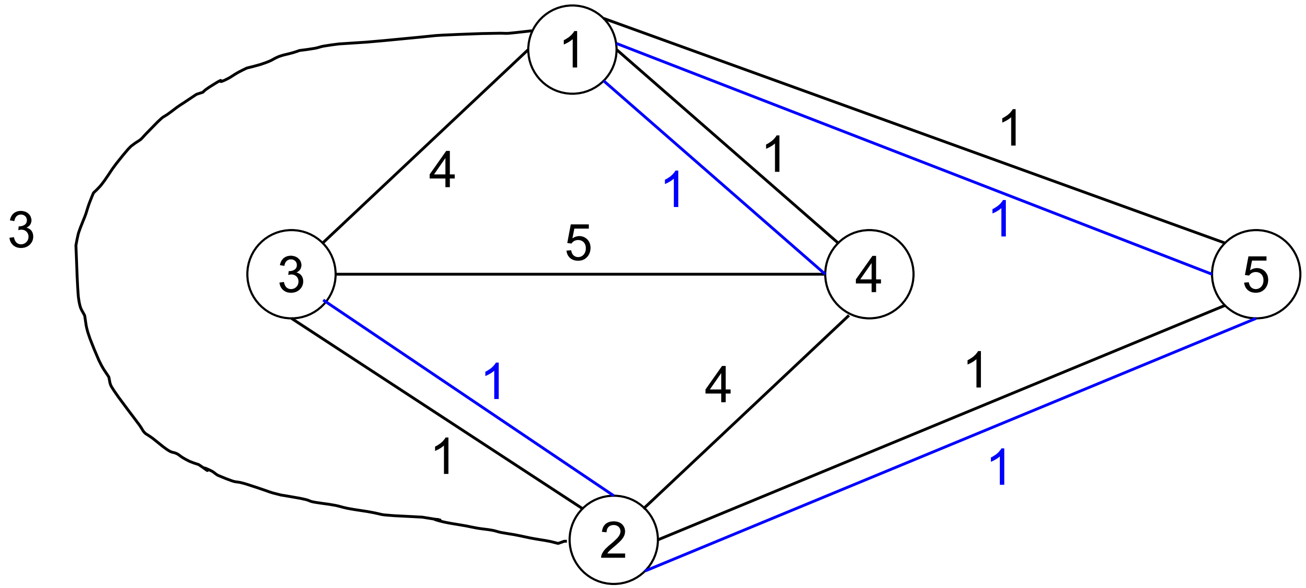
Ação: duplicar as arestas

Exemplo



O grafo G* é um grafo eureliano?

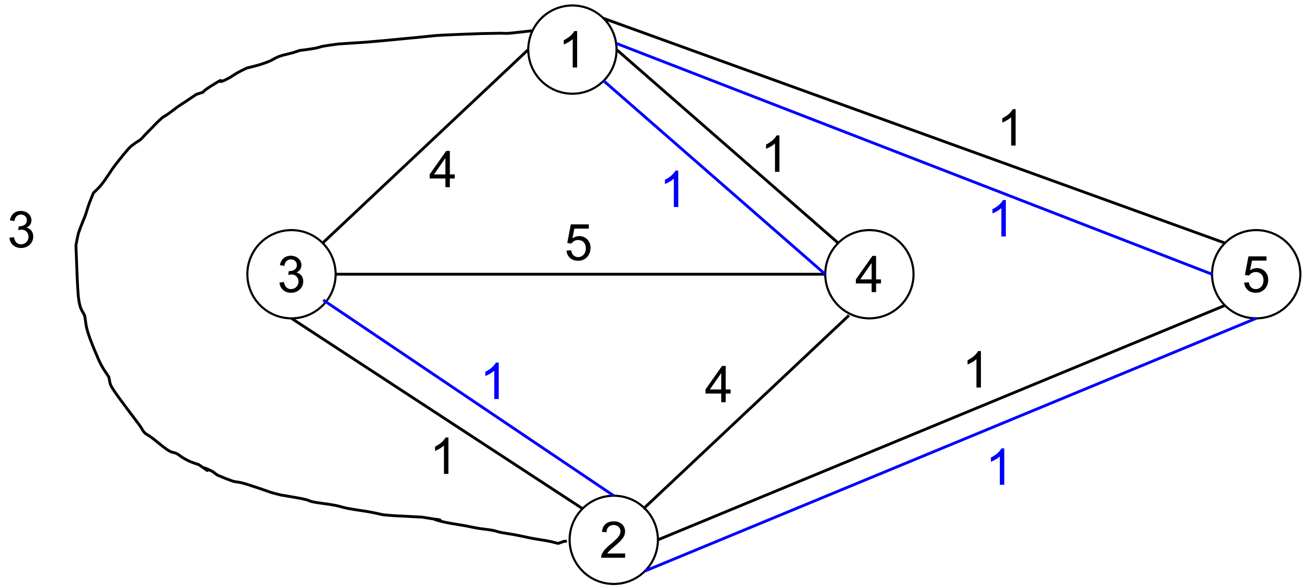
Exemplo



O grafo G* é um grafo eureliano? sim, desde que todos os vértices são pares

Ação: aplicar um algoritmo de determinação de ciclo eureliano

Exemplo



Existem várias possibilidades, mas qualquer ciclo eureliano será de peso mínimo

Definições

- Grafo Hamiltoniano
  - é possível visitar todos os vértices, porém passando em cada vértice apenas uma vez
  - o vértice origem e o vértice destino são os mesmos (forma-se um ciclo hamiltoniano)

Definições

- Caminho hamiltoniano
  – caminho que inclui cada vértice do grafo exatamente uma vez
- Ciclo hamiltoniano
  – ciclo que inclui cada vértice do grafo exatamente uma única vez, repetindo apenas o vértice de partida

Classe de Problemas

- P
  - determinar se um grafo possui um ciclo euleriano

X

- NP
  - determinar se um grafo possui um ciclo hamiltoniano

↓

não determinístico em tempo polinomial
NP completo

Problema do Caixeiro Viajante

- Um dos problemas mais estudados na otimização da teoria de grafos
  - problema clássico de otimização combinatória
- Dado um grafo G = (V, A) ponderado (direcionado ou não)
  - encontre um ciclo hamiltoniano de peso mínimo
- Problema NP difícil

Exemplos de Aplicação

- Visita a diferentes cidades
  - um caixeiro viajante deseja visitar várias cidades e voltar para o ponto de origem, de forma que ele visite cada cidade exatamente uma vez e percorra a menor distância
- Outros exemplos
  - sequenciamento de itens em uma máquina
  - sequenciamento de genoma
  - gerenciamento de frotas de veículos para carga e descarga de bagagens de aeronaves

Solução

Se o grafo é hamiltoniano

então resolver o problema

senão transformar o grafo em um grafo completo

// grafo simples em que cada vértice está

// conectado a todos os outros

resolver o problema

Transformação do Grafo

- Grafo G é transformado em um grafo completo
  - se a aresta existe

    então ela permanece inalterada no grafo

    senão cria a aresta de forma que ela

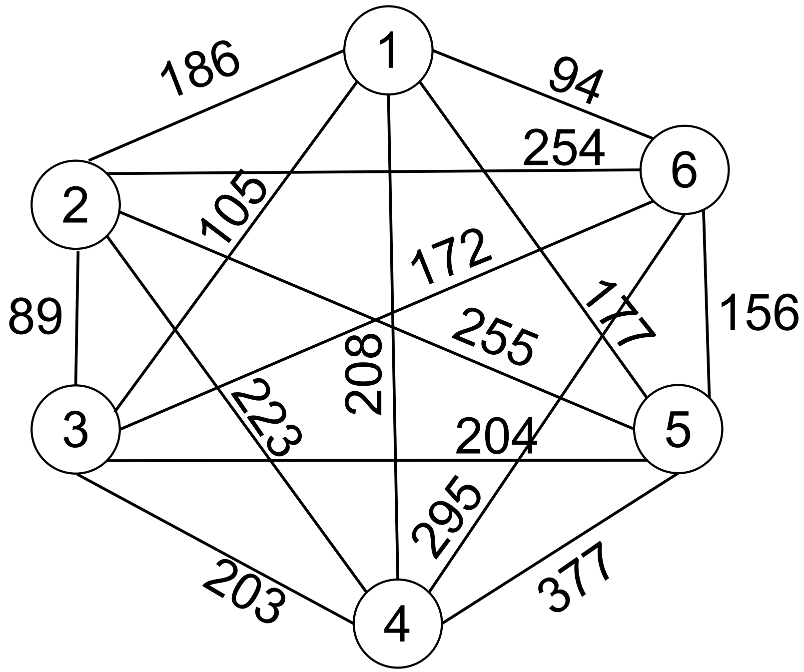
    possua um peso muito grande

    ex.: o peso da aresta é a soma dos pesos de todas as arestas do grafo

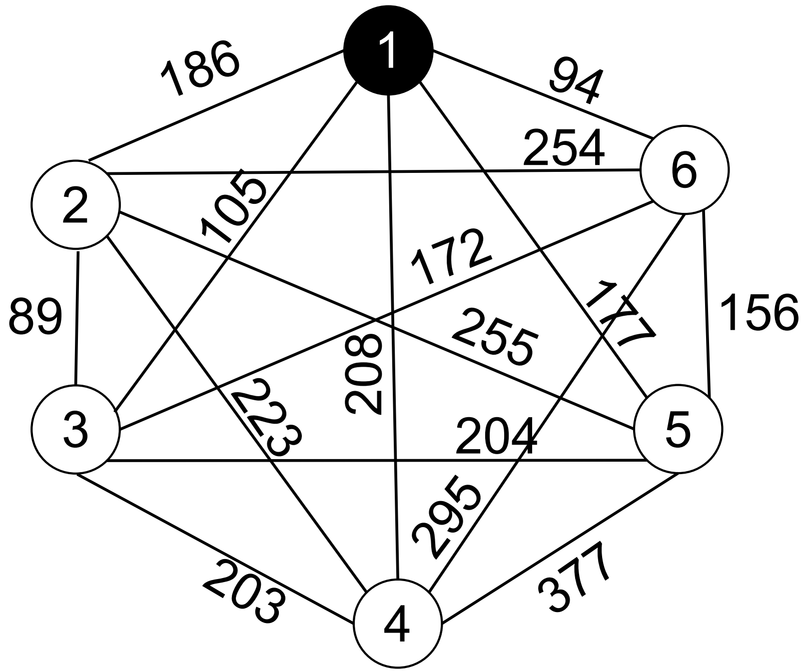
Resolver o Problema

- Modelo de programação inteira
  - existem diversos modelos matemáticos
  - para problemas de grande porte encontrados na prática, o uso dos modelos pode não ser eficaz
- Heurísticas
  - existem diversas heurísticas

Heurística: Vizinho mais Próximo

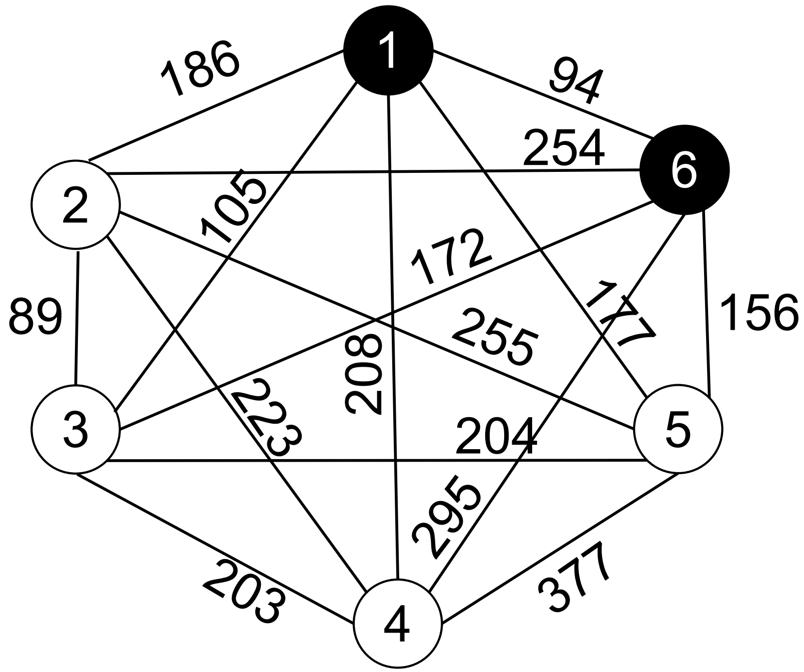


Heurística: Vizinho mais Próximo



vértice i: 1
rota: 1
distância: 0
vértice j mais próximo de i que ainda não foi visitado: 6

Heurística: Vizinho mais Próximo



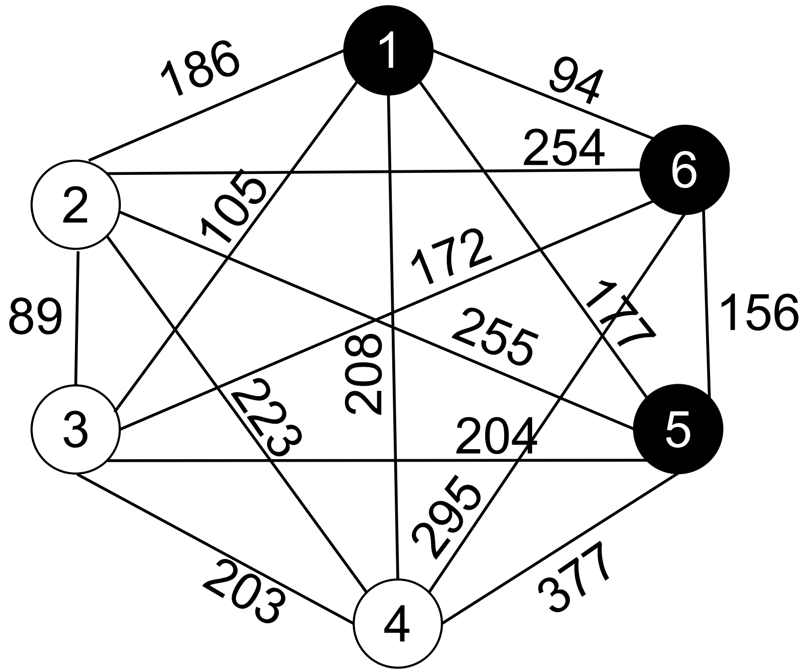
vértice i: 6

rota: 1 6

distância: 0 + 94

vértice j mais próximo de i que ainda não foi visitado: 5

Heurística: Vizinho mais Próximo



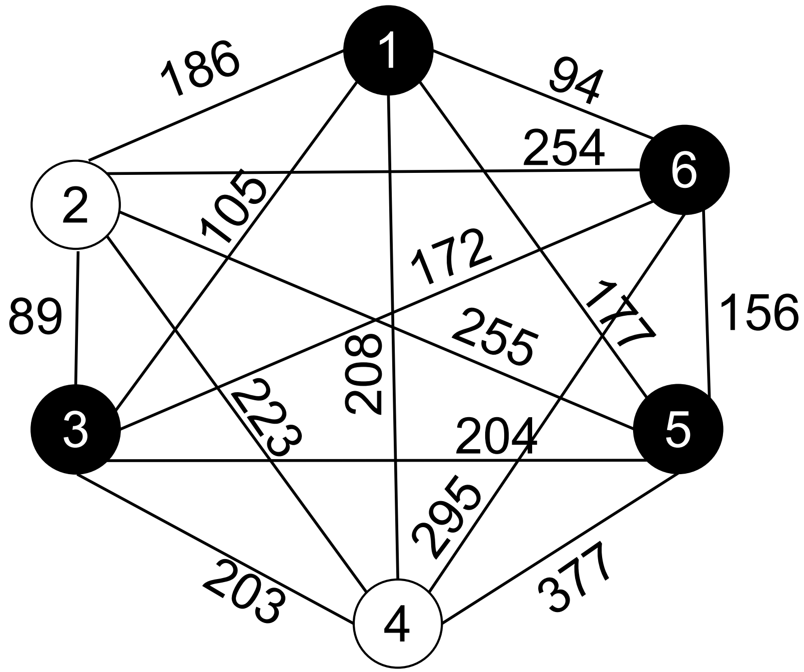
vértice i: 5

rota: 1 6 5

distância: 0 + 94 + 156

vértice j mais próximo de i que ainda não foi visitado: 3

Heurística: Vizinho mais Próximo



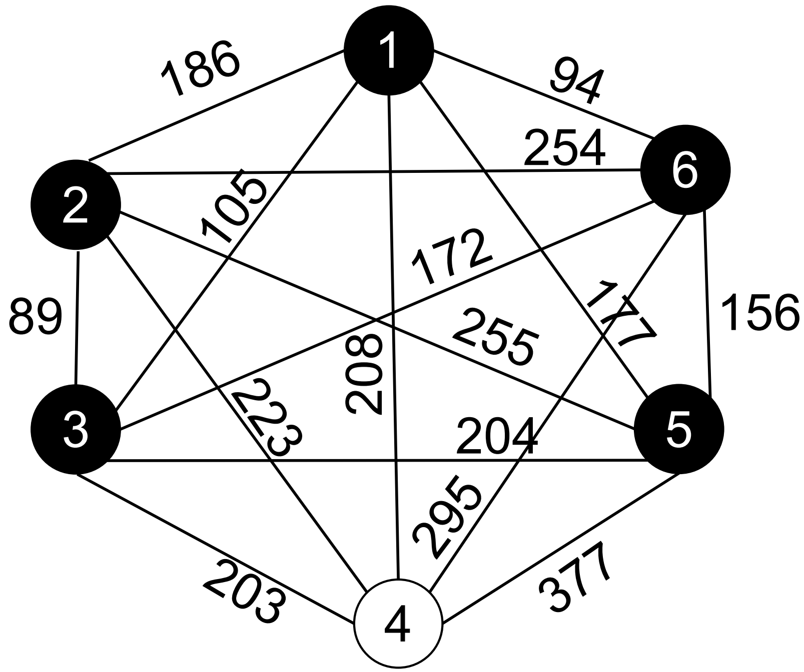
vértice i: 3

rota: 1 6 5 3

distância: 0 + 94 + 156 + 204

vértice j mais próximo de i que ainda não foi visitado: 2

Heurística: Vizinho mais Próximo



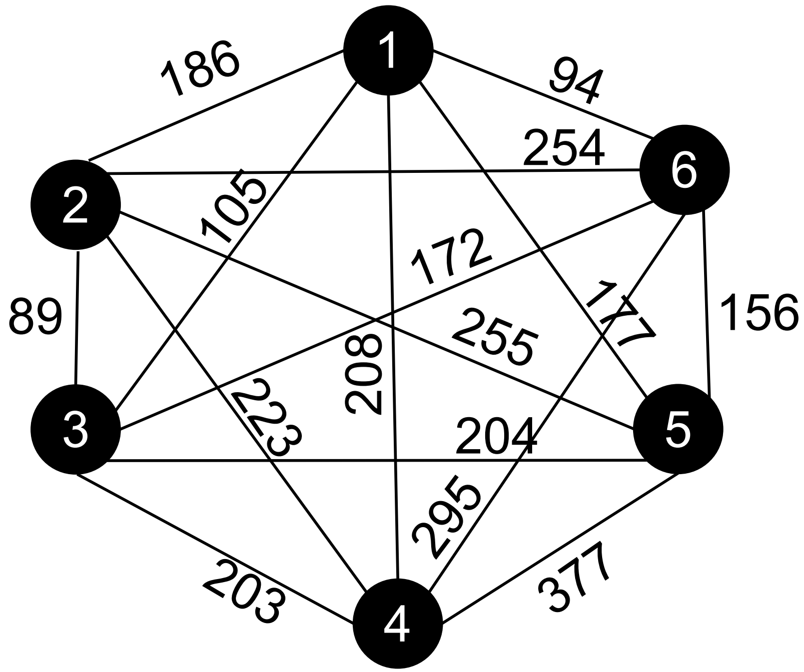
vértice i: 2

rota: 1 6 5 3 2

distância: 0 + 94 + 156 + 204 + 89

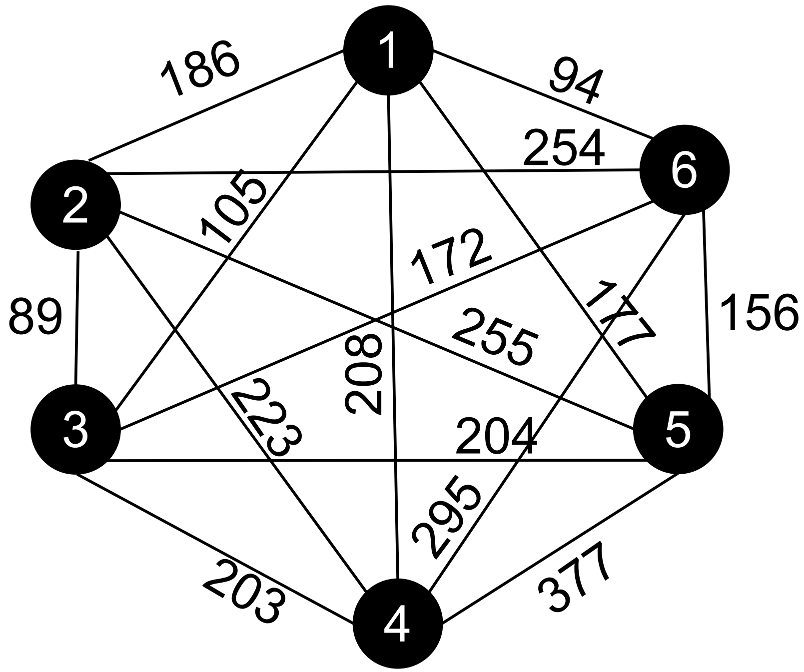
vértice j mais próximo de i que ainda não foi visitado: 4

Heurística: Vizinho mais Próximo



vértice i: 4
rota: 1 6 5 3 2 4
distância: 0 + 94 + 156 + 204 + 89 + 223
vértice j mais próximo de i que ainda não foi visitado: -

Heurística: Vizinho mais Próximo



vértice i: -

rota: 1 6 5 3 2 4 1 // acrescenta o vértice de origem

distância: 0 + 94 + 156 + 204 + 89 + 223 + 208 // acrescenta a distância
= 974