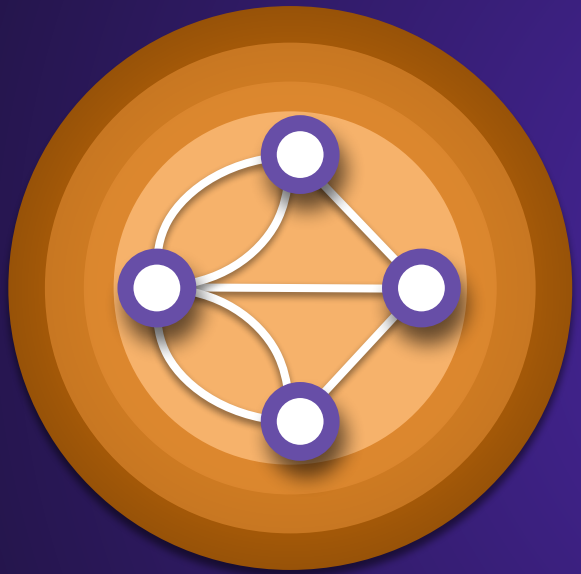


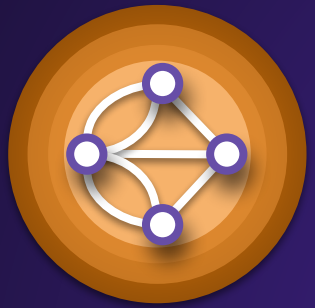
BIG GRAPH

BIG DATA APLICADO A GRAFOS GIGANTES E DINÂMICOS

@juanplopes

11/maio/2018

QCon SÃO PAULO



VANITY SLIDE

Acho que é obrigatório, né?

JUAN LOPES, PAPAI DO MIGUEL

@juanplopes

R&D SOFTWARE ENGINEER

INTELIE (a **RigNet** company)

DOUTORANDO EM ALGORITMOS

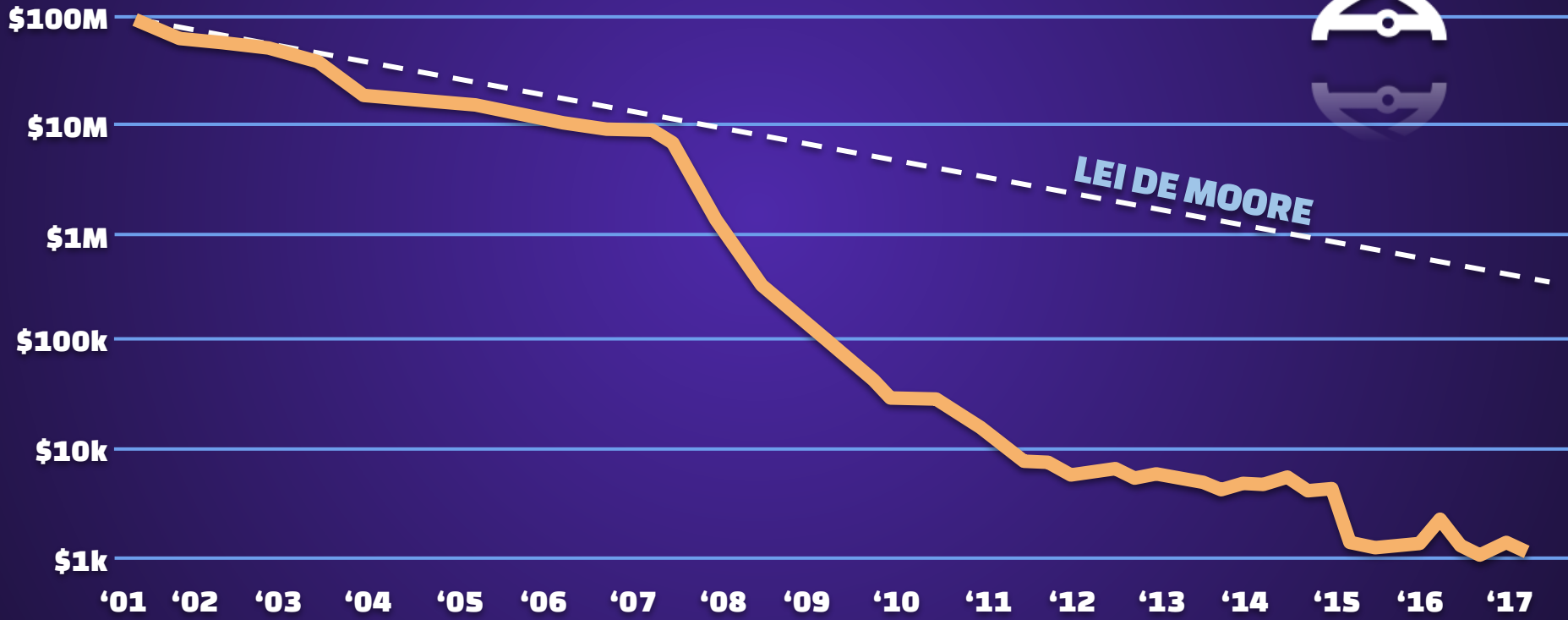
COPPE/UFRJ


PROJETO GENOMA
HUMANO
1990-2001

O Projeto Genoma Humano custou cerca de US$ 3 bilhões ao longo de 11 anos para sequenciar completamente o DNA humano.

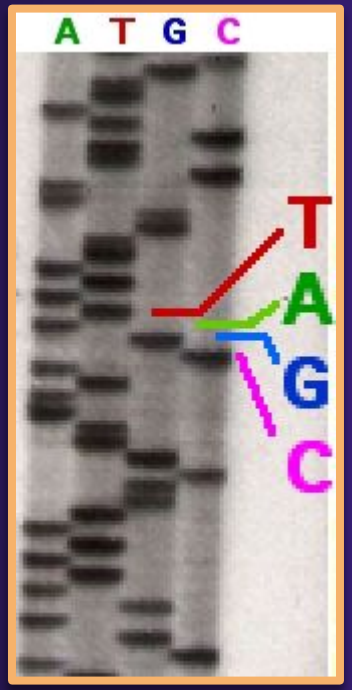
CUSTO DE SEQUENCIAMENTO

Genoma Humano



https://www.genome.gov/27541954/dna-sequencing-costs-data/

Atualmente, é possível realizar sequenciamento parcial do próprio DNA por até US$ 99.



Shotgun Sequencing é um método de sequenciamento que **quebra** a cadeia em **pontos aleatórios**, fáceis de ler, e depois reagrupa as leituras **computacionalmente**.

O sequenciador quebra a cadeia em vários pontos aleatórios.

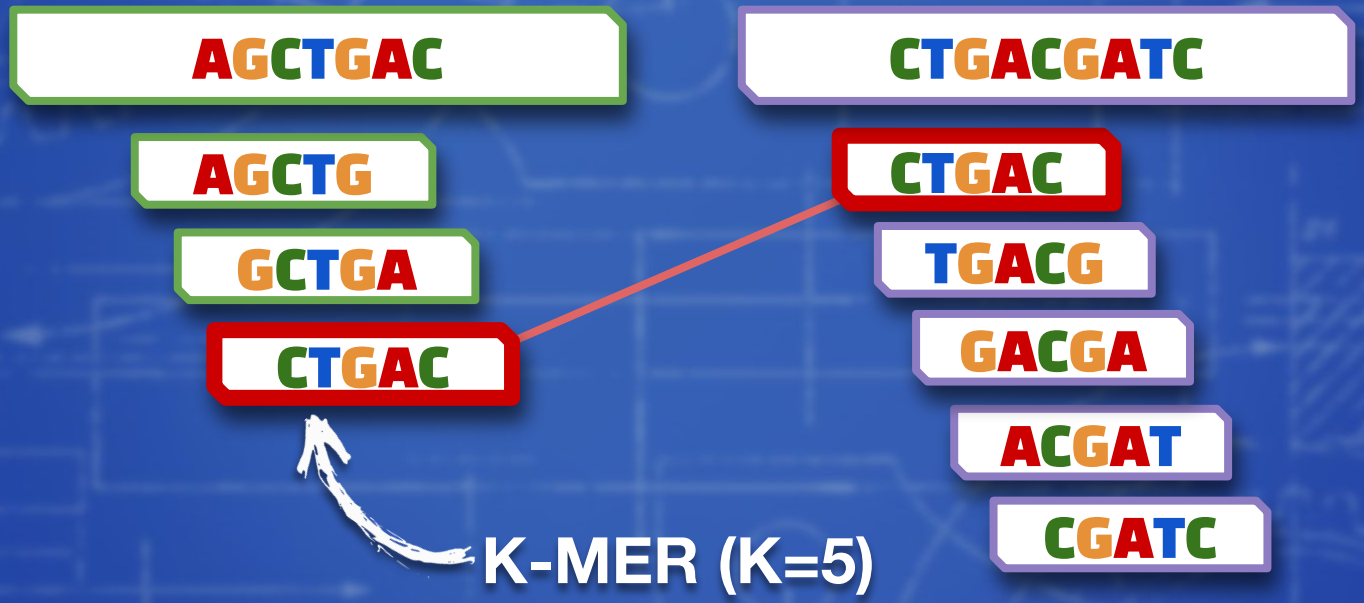
...AGCTGACGATCGGAAGATCAG...

REMONTAGEM (*DE NOVO* ASSEMBLY)

É preciso remontar a sequência original

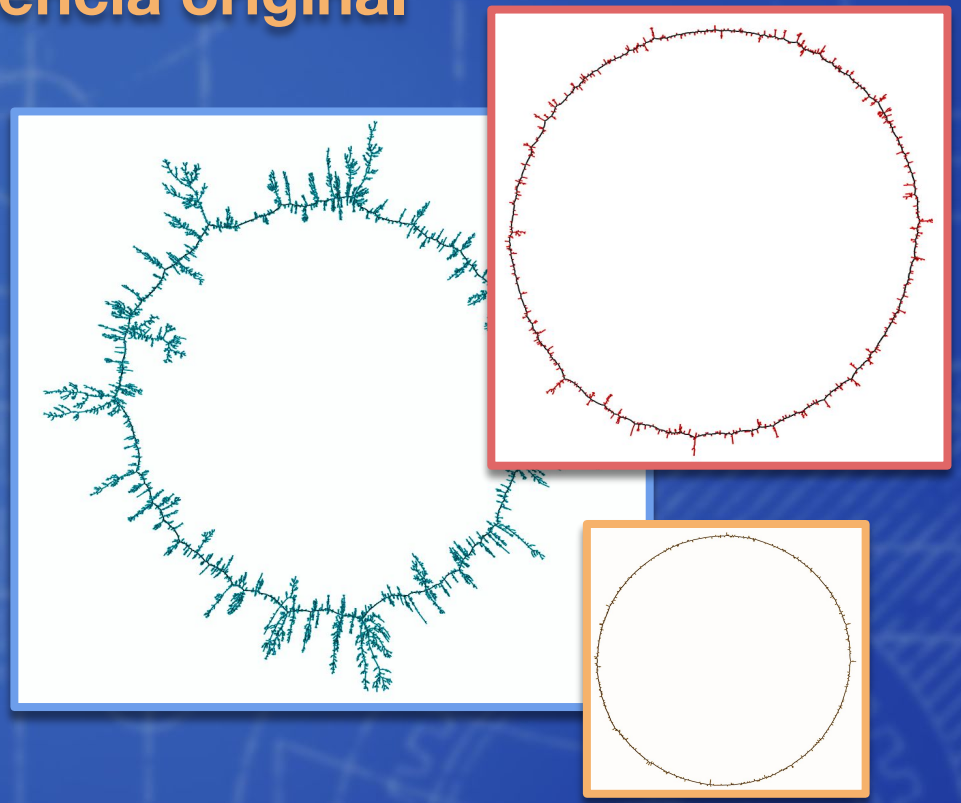
REMONTAGEM (*DE NOVO* ASSEMBLY)

É preciso remontar a sequência original



REMONTAGEM (*DE NOVO* ASSEMBLY)

É preciso remontar a sequência original



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3421212/

Drosófila

123 milhões de pares (22GB)

Humano

3.3 bilhões de pares (600GB)

Psilotum nudum

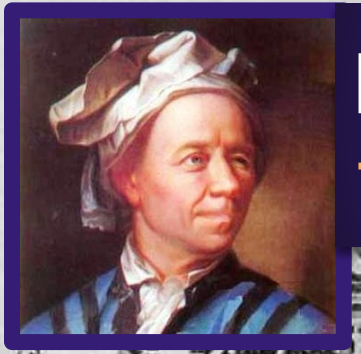
250 bilhões de pares (45TB)

http://www.biology-pages.info/G/GenomeSizes.html
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5411771/

MAS AFINAL,

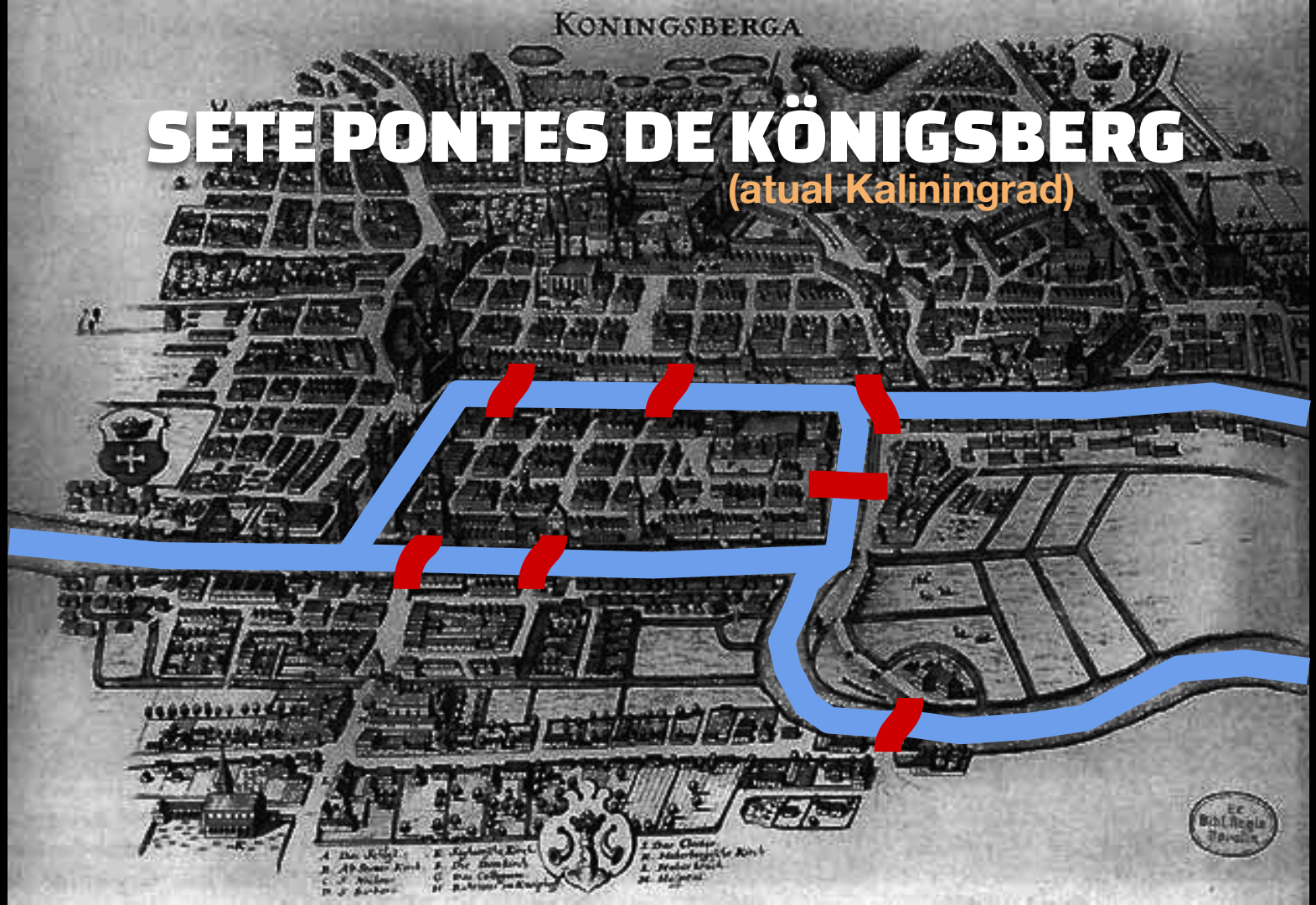
O QUE SÃO GRAFOS?!

Você quis dizer: ***gráficos***?



LEONHARD EULER

1707-1783


KONINGSBERGA
SETE PONTES DE KÖNIGSBERG
(atual Kaliningrad)

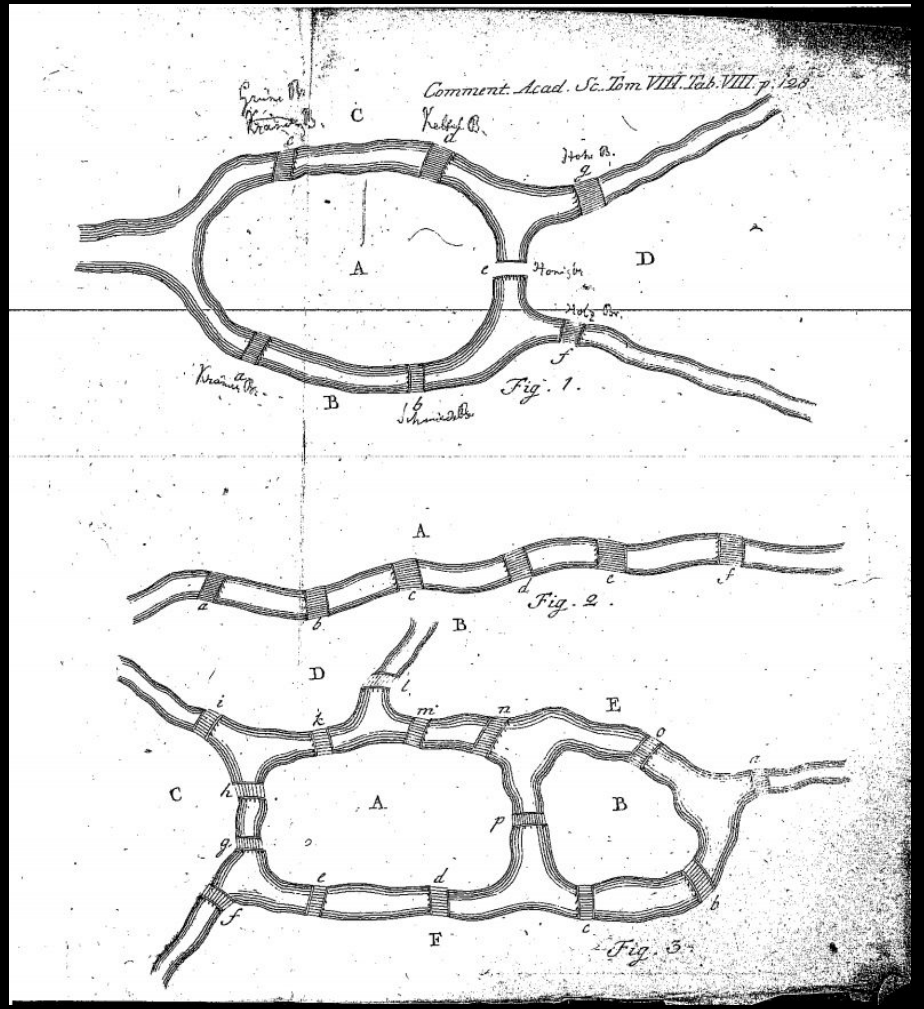

KONINGSBERGA
SETE PONTES DE KÖNIGSBERG
(atual Kaliningrad)

SOLVTIO PROBLEMATIS AD GEOMETRIAM SITVS PERTINENTIS.

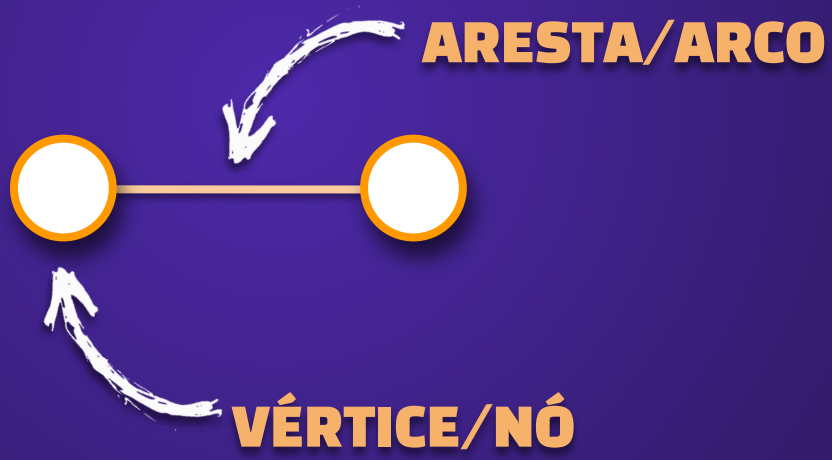
AVCTORE
Leonh. Eulero.

§. 1.

Tabula VIII. PRaeter illam Geometriae partem, quae circa quantitates verſatur, et omni tempore ſummo ſtudio eſt exculta, alterius partis etiamnum admodum ignotae primus mentionem fecit *Leibnitzius*, quam Geometriam ſitus vocauit. Iſta pars ab ipſo in ſolo ſitu determinando, ſitusque proprietatibus eruendis occupata eſſe ſtatuitur; in quo negotio neque ad quantitates reſpiciendum, neque calculo quantitatum vtendum ſit. Cuiusmodi autem problemata ad hanc ſitus Geometriam pertineant, et quali methodo in iis reſoluendis vti oporteat, non ſatis eſt definitum. Quamobrem, cum nuper problematis cuiusdam mentio eſſet facta, quod quidem ad geometriam pertinere videbatur, at ita erat comparatum, vt neque determinationem quantitatum requireret, neque ſolutionem calculi quantitatum ope admitteret, id ad geometriam ſitus referre haud dubitaui: praeſertim quod in eius ſolutione ſolus ſitus in conſiderationem venat, calculus vero nullius prorſus ſit vſus. Methodum ergo meam quam ad huius generis problemata

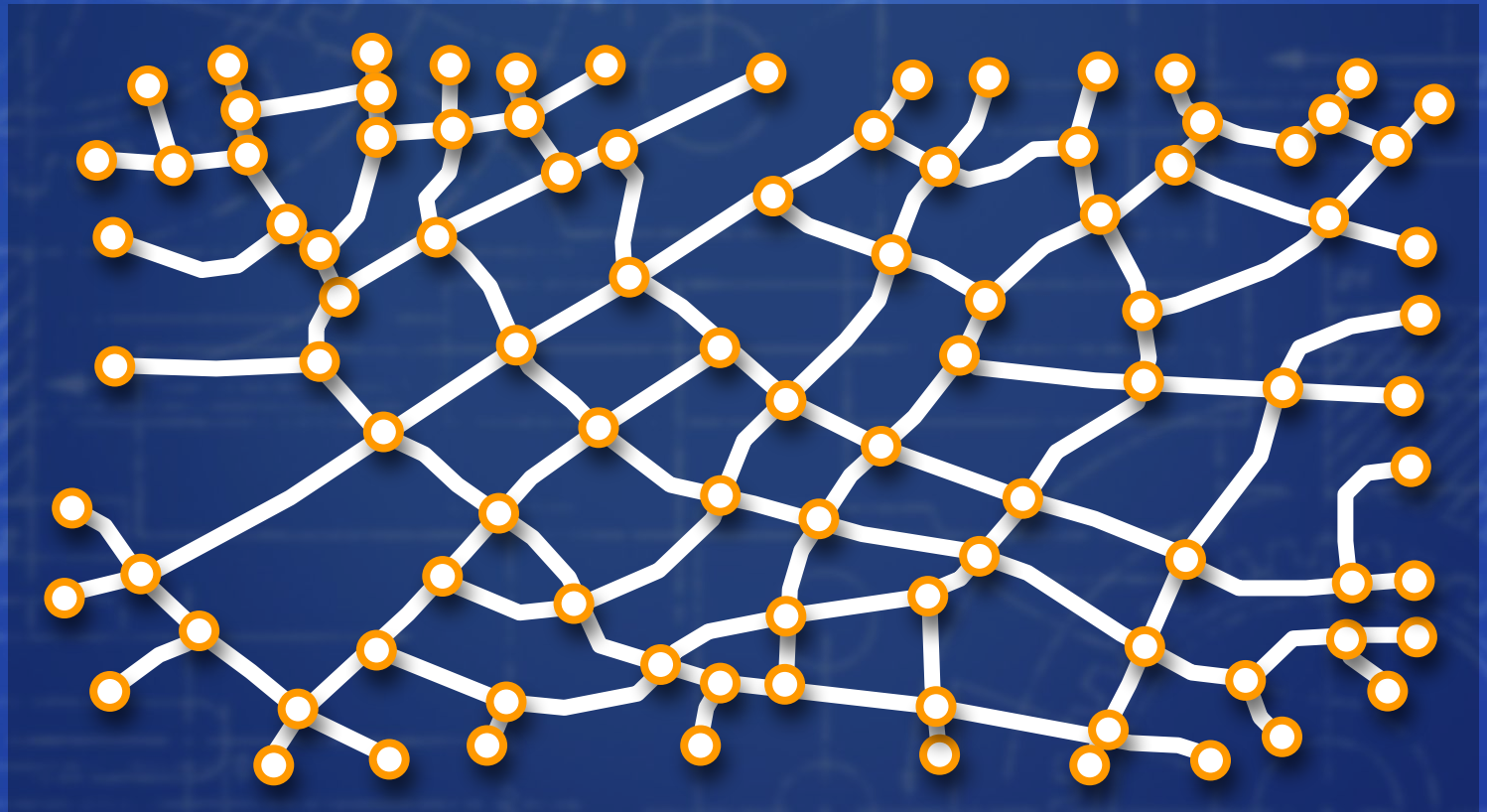


Grafos modelam relações.



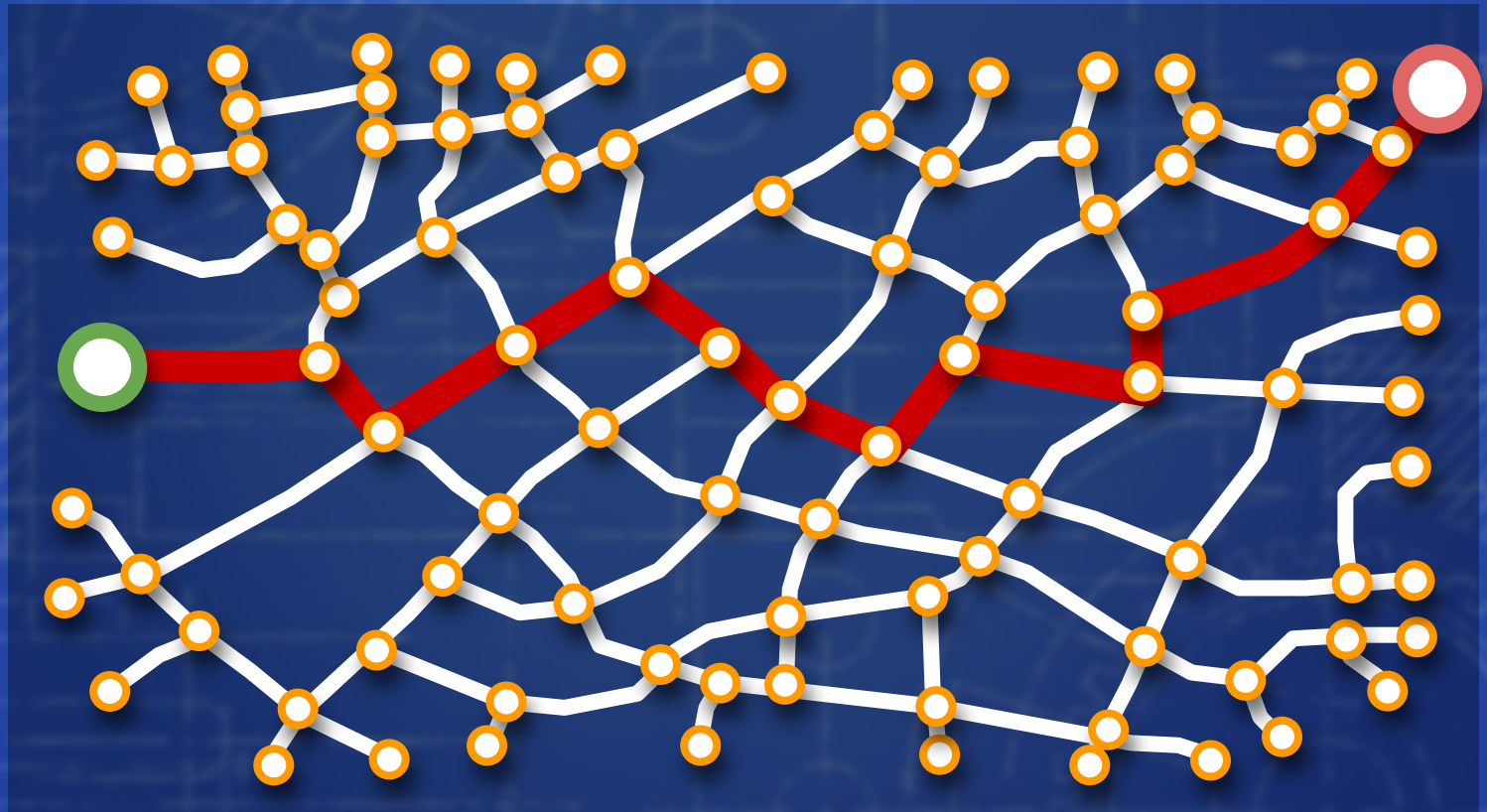
PODEM MODELAR CIDADES

Cada vértice representa a interseção de duas ruas



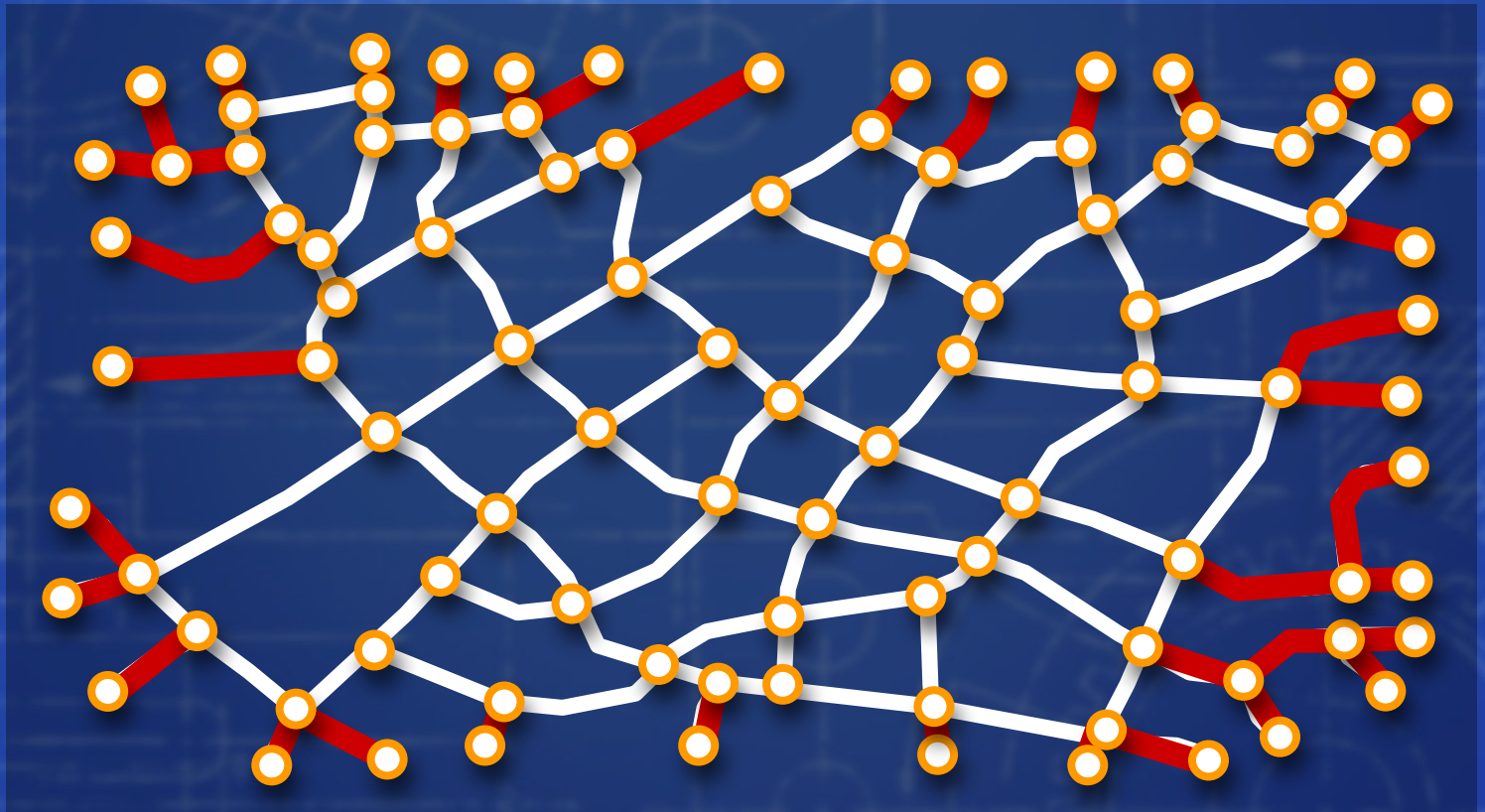
CAMINHO MAIS CURTO

Algoritmos: Dijkstra, Bellman-Ford, Floyd-Warshall, etc...



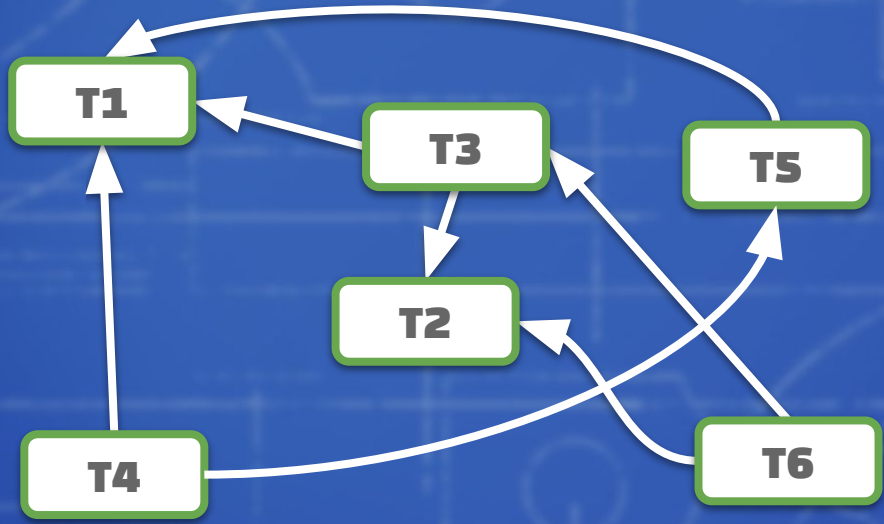
DETECÇÃO DE PONTES

Algoritmo: Tarjan



... SEQUÊNCIAS DE TAREFAS

Cada aresta (dirigida) representa uma dependência

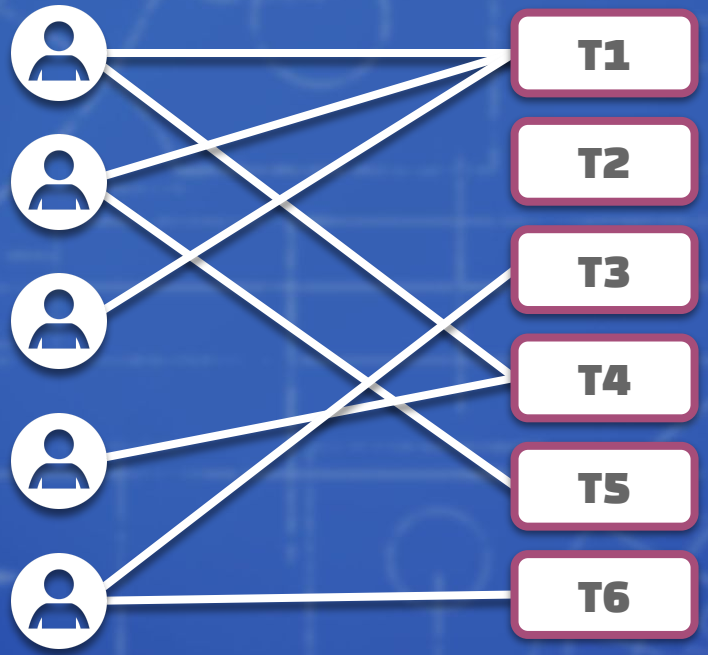


ORDENAÇÃO TOPOLÓGICA

Algoritmo: Khan

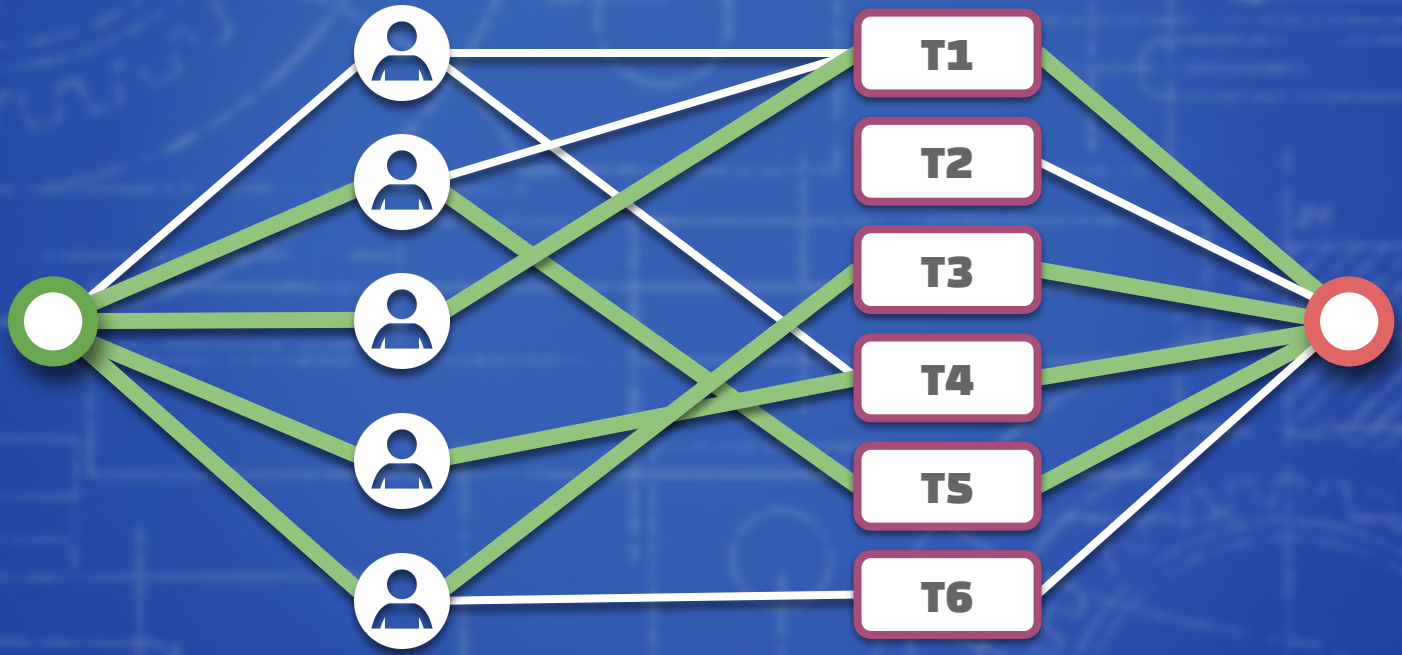
... HABILIDADES VS TAREFAS

Cada aresta representa habilidade de realizar uma tarefa



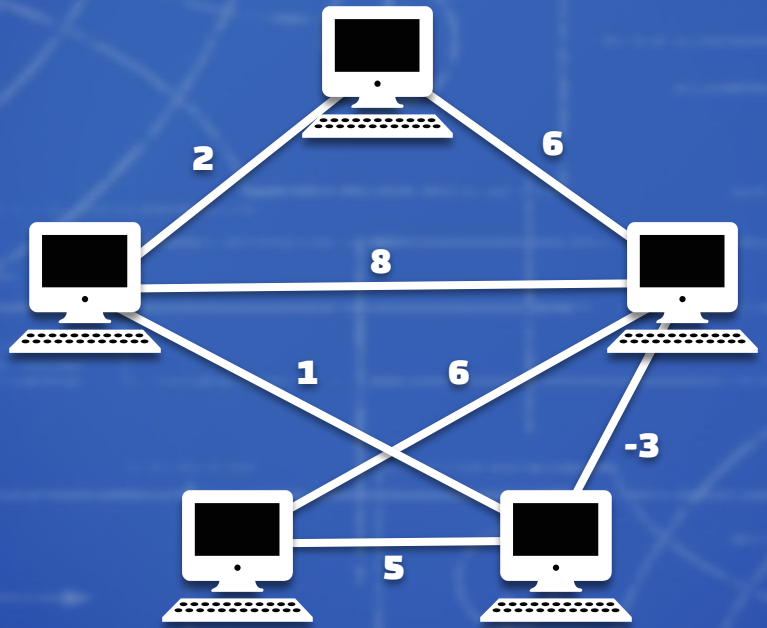
EMPARELHAMENTO MÁXIMO

Algoritmos: Ford-Fulkerson, Edmonds-Karp, Dinic, etc...



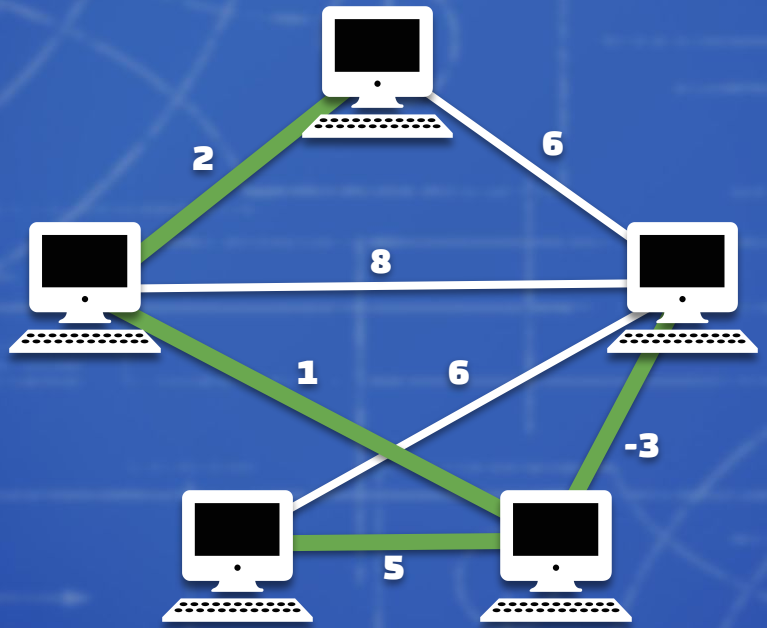
... REDES (DIVERSAS)

Cada aresta representa o custo de conexão



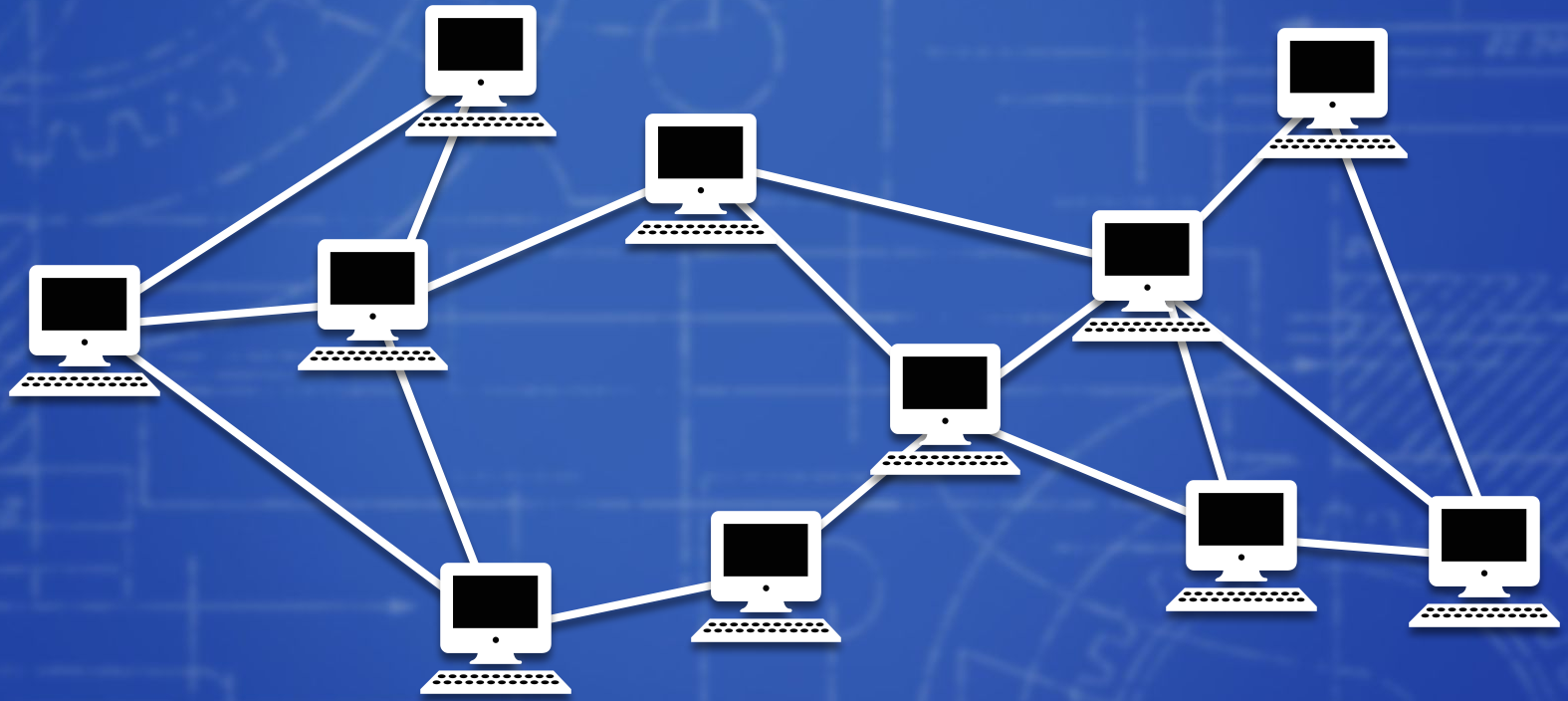
ÁRVORE GERADORA MÍNIMA

Algoritmos: Prim, Kruskal, Chazelle, etc...



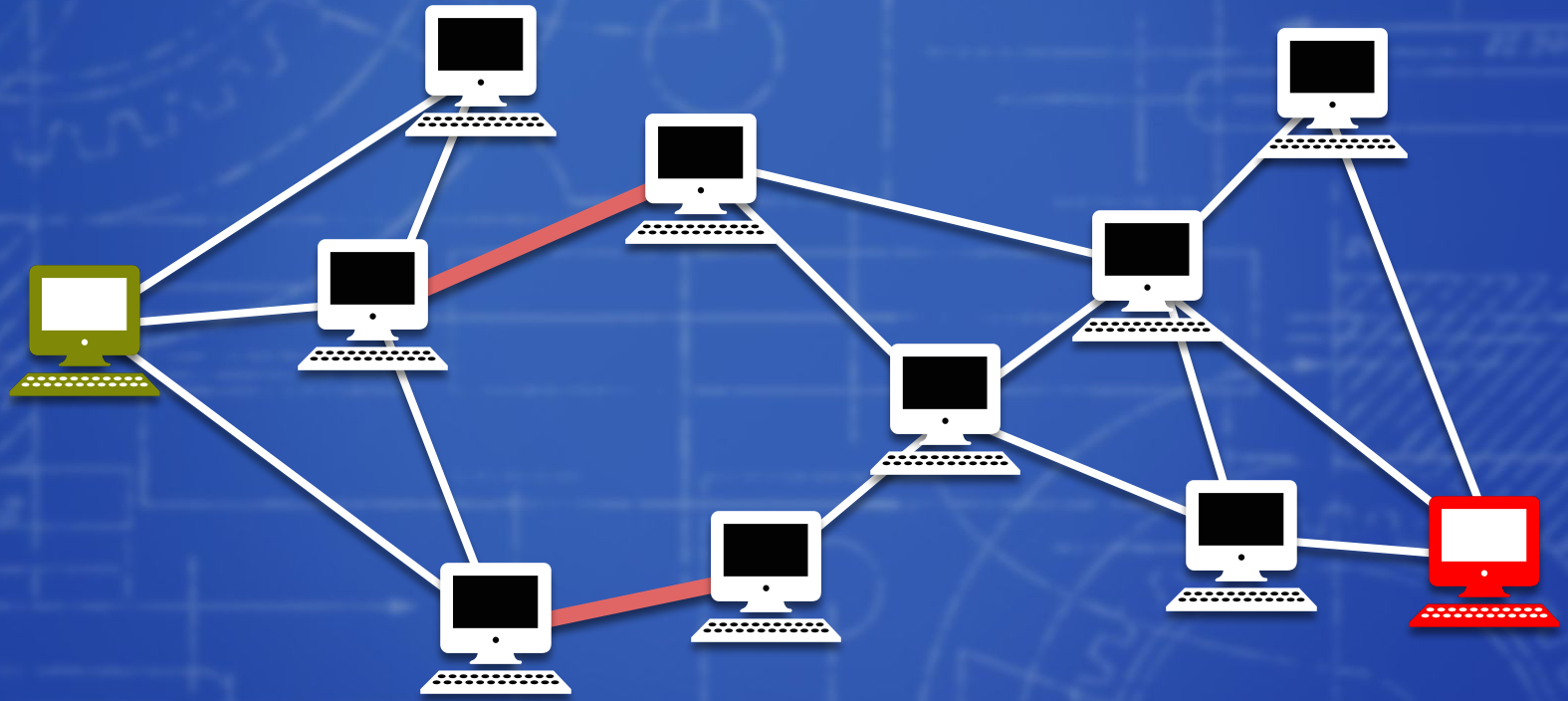
... REDES (2)

Cada aresta representa um link



CORTE MÍNIMO S-T (max-flow min-cut theorem)

Algoritmos: Ford-Fulkerson, Edmonds-Karp, Dinic, etc...



São muitos algoritmos!

E continuam surgindo novos.

Só se fala de grafos em computação.

ENCONTRO DE TEORIA DA COMPUTAÇÃO

CSBC 2017: Somente os 20 primeiros papers

- Deletion Graph Problems Based on Deadlock Resolution
- Número de Ramsey relativo a arestas de potências de caminhos
- Uma Aproximação para o Problema de Alocação de Terminais com Capacidade
- Representações Implícitas Probabilísticas de Grafos
- Vertex-disjoint path covers in graphs
- Biclique edge-choosability in some classes of graphs
- Facility Leasing with Penalties
- Advances in Anti-Ramsey Theory for random graphs
- Método Exato para um Problema de Alocação Justa
- Hitting all longest cycles in a graph
- Coloração arco-íris em grafos resultantes de produto cartesiano
- Um algoritmo exato para biclique-coloração
- Edge-colouring of triangle-free graphs with no proper majors
- Modelos para o Problema de Alocação de Pedágios
- Uma versão algoritmica do Lema Local de Lovász
- Análise dos Tempos de Setup Dependentes da Sequência em...
- New Insights on Prize Collecting Path Problem
- Tight bounds for gap-labellings
- Grafos do tipo Half Cut
- The 1,2,3-Conjecture for powers of paths and powers of cycles

http://csbc2017.mackenzie.br/eventos/2-etc

ENCONTRO DE TEORIA DA COMPUTAÇÃO

CSBC 2017: Somente os 20 primeiros papers

- Deletion Graph Problems Based on Deadlock Resolution
- Número de Ramsey relativo a arestas de potências de caminhos
- Uma Aproximação para o Problema de Alocação de Terminais com Capacidade
- Representações Implícitas Probabilísticas de Grafos
- Vertex-disjoint path covers in graphs
- Biclique edge-choosability in some classes of graphs
- Facility Leasing with Penalties
- Advances in Anti-Ramsey Theory for random graphs
- Método Exato para um Problema de Alocação Justa
- Hitting all longest cycles in a graph
- Coloração arco-íris em grafos resultantes de produto cartesiano
- Um algoritmo exato para biclique-coloração
- Edge-colouring of triangle-free graphs with no proper majors
- Modelos para o Problema de Alocação de Pedágios
- Uma versão algoritmica do Lema Local de Lovász
- Análise dos Tempos de Setup Dependentes da Sequência em...
- New Insights on Prize Collecting Path Problem
- Tight bounds for gap-labellings
- Grafos do tipo Half Cut
- The 1,2,3-Conjecture for powers of paths and powers of cycles

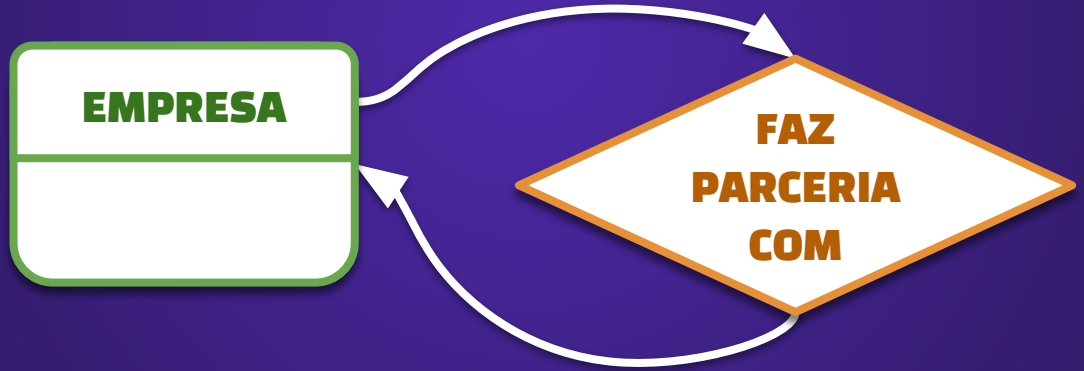
Em laranja os que são diretamente sobre grafos.

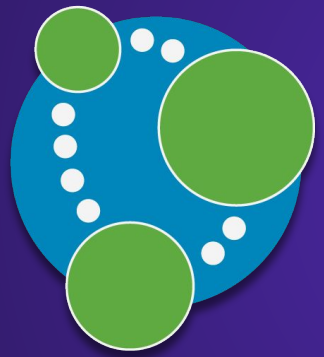
Em vermelho os que usam teoria de grafos para resolver algum problema.

http://csbc2017.mackenzie.br/eventos/2-etc

Se a academia fala tanto de grafos, onde eles estão nas aplicações comerciais?

Mais perto do que você imagina.





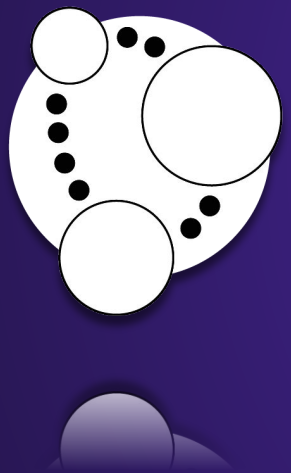
neo4j

CYPHER

Neo4j declarative query language

```
MATCH (actor:Person)-[:ACTED_IN]->(movie:Movie)
WHERE movie.title STARTS WITH "T"
RETURN movie.title AS title, collect(actor.name) AS cast
ORDER BY title ASC LIMIT 10;
```

https://neo4j.com/blog/graph-compute-neo4j-algorithms-spark-extensions/



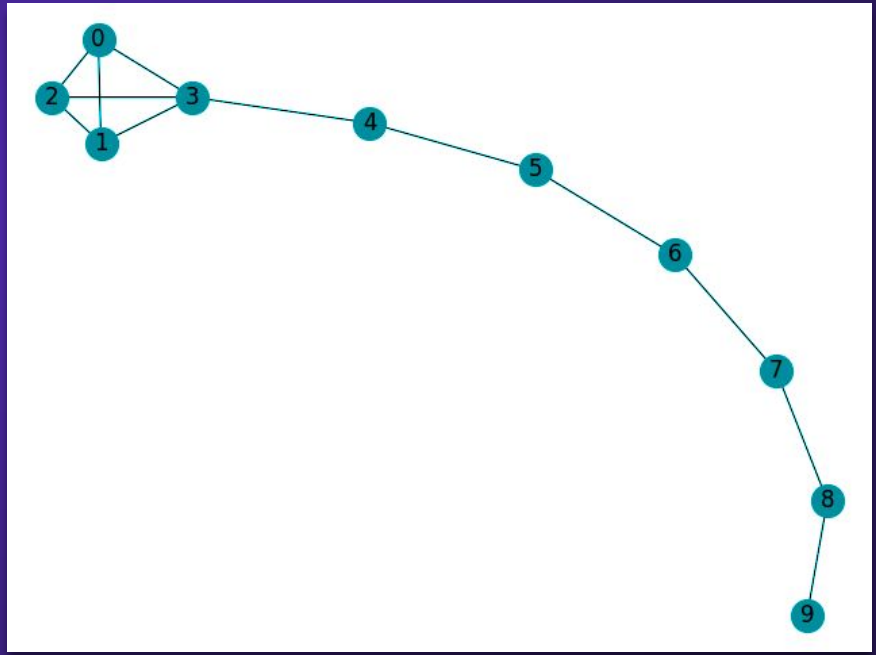
“Neo4j is optimized for online transaction processing (OLTP) and is intended to be used as your primary database. While it wasn’t built specifically with the intention of being used for graph compute or analytics, a lot of customers and open-source users are using Neo4j for those purposes.”

https://neo4j.com/blog/graph-compute-neo4j-algorithms-spark-extensions/

NETWORKX

Biblioteca em Python para algoritmos em grafos

```
#    Copyright (C) 2004-2018 by
#    Aric Hagberg <hagberg@lanl.gov>
#    Dan Schult <dschult@colgate.edu>
#    Pieter Swart <swart@lanl.gov>
#    All rights reserved.
#    BSD license.

import matplotlib.pyplot as plt
from networkx import nx

G = nx.lollipop_graph(4, 6)

pathlengths = []

print("source vertex {target:length, }")
for v in G.nodes():
    spl = dict(nx.single_source_shortest_path_length(G, v))
    print('{} {} '.format(v, spl))
    for p in spl:
        pathlengths.append(spl[p])

print('')
print("average shortest path length %s" % (sum(pathlengths) / len(pathlengths)))

# histogram of path lengths
dist = {}
for p in pathlengths:
    if p in dist:
        dist[p] += 1
    else:
        dist[p] = 1

print('')
print("length #paths")
verts = dist.keys()
for d in sorted(verts):
    print('%s %d' % (d, dist[d]))

print("radius: %d" % nx.radius(G))
print("diameter: %d" % nx.diameter(G))
print("eccentricity: %s" % nx.eccentricity(G))
print("center: %s" % nx.center(G))
print("periphery: %s" % nx.periphery(G))
print("density: %s" % nx.density(G))

nx.draw(G, with_labels=True)
plt.show()
```

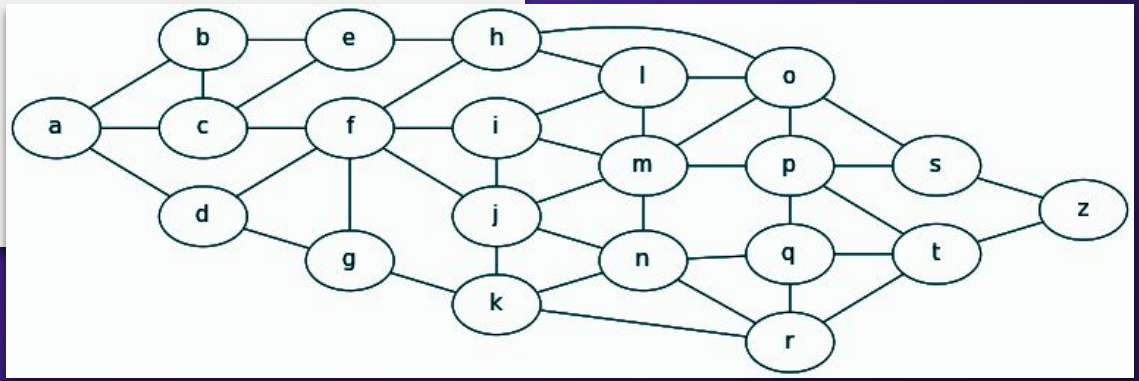


https://networkx.github.io/documentation/stable/auto_examples/basic/plot_properties.html

GRAPHVIZ

Ferramenta para layout visual de grafos

```
graph {
    rankdir=LR;
    a -- { b c d }; b -- { c e }; c -- { e f }; d -- { f g }; e -- h;
    f -- { h i j g }; g -- k; h -- { o l }; i -- { l m j }; j -- { m n k };
    k -- { n r }; l -- { o m }; m -- { o p n }; n -- { q r };
    o -- { s p }; p -- { s t q }; q -- { t r }; r -- t; s -- z; t -- z;
    { rank=same; b, c, d }
    { rank=same; e, f, g }
    { rank=same; h, i, j, k }
    { rank=same; l, m, n }
    { rank=same; o, p, q, r }
    { rank=same; s, t }
}
```



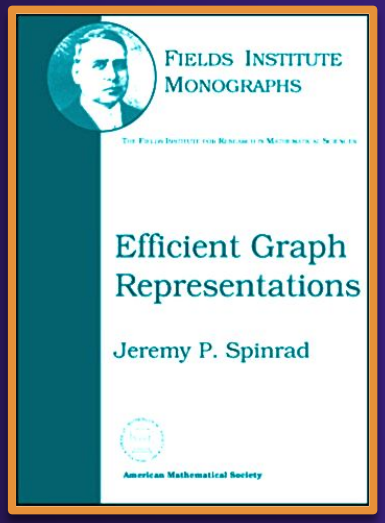
https://graphs.grevian.org/example

Grafos em tempos de BigData são

GIGANTES e DINÂMICOS

Bilhões de vértices.
Trilhões de arestas.
$m = O(n^2)$.

Inserção e remoção
de vértices e arestas.
Poucos algoritmos.



Parte do problema de lidar com grafos gigantes é simplesmente uma questão de representação.

https://bookstore.ams.org/fim-19/

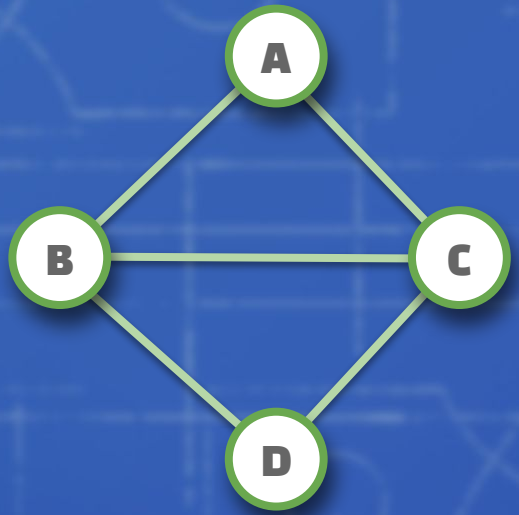
REPRESENTAÇÕES DE GRAFOS

Cada uma tem vantagens e desvantagens

Matriz de adjacência

	A	B	C	D
A	0	1	1	0
B	1	0	1	1
C	1	1	0	1
D	0	1	1	0

$O(n^2)$ bits



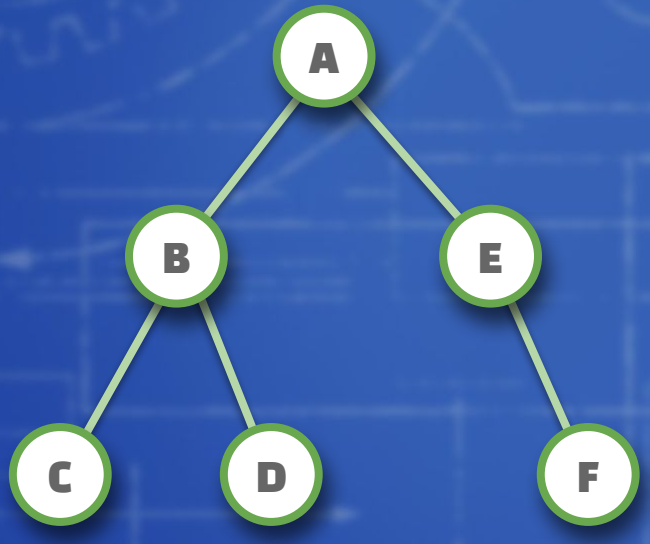
Lista de adjacência

A	B	C	
B	A	C	D
C	A	B	C
D	B	C	

$O(m \log n)$ bits

REPRESENTAÇÕES DE GRAFOS

Classes diferentes podem ter diferentes representações ótimas



A	B	C	D	E	F
	A	B	B	A	E

O(n log n) bits

GRAFOS DE DEBRUIJN

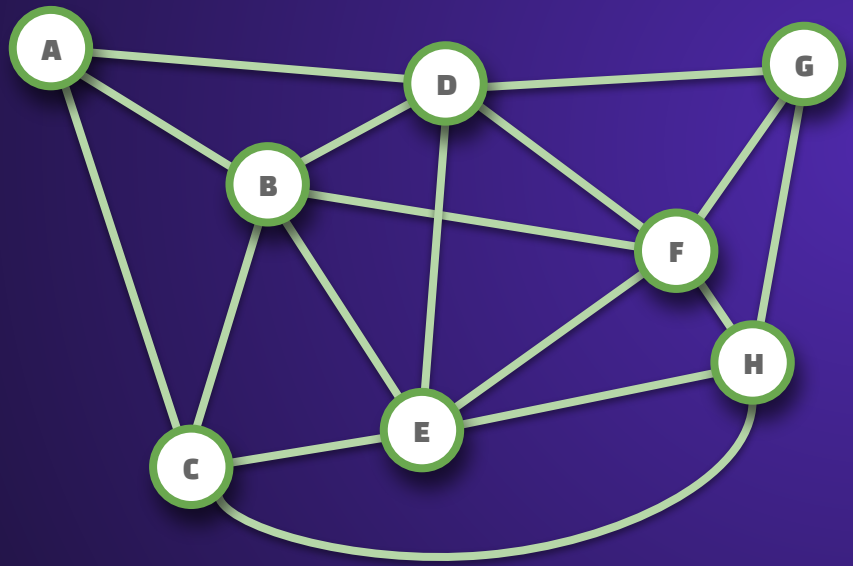
Permitem remontar genoma a partir de fragmentos

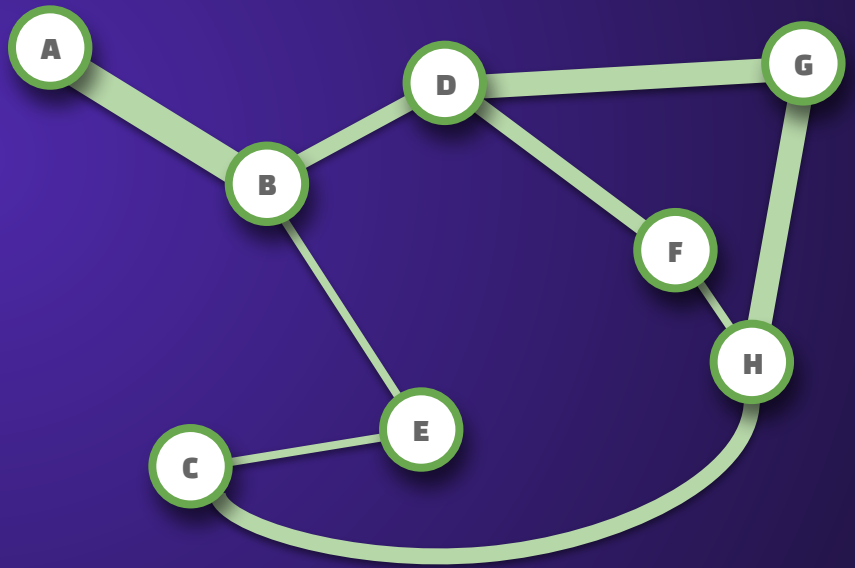
"We relate changes in local and global graph connectivity to the false positive rate of the underlying Bloom filters and show that the graph's global structure is accurate for false positive rates of 15% or lower, corresponding to a lower memory limit of approximately 4 bits per graph node."

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3421212/

SPANNERS E SPARSIFIERS

Aproximam grafos densos através de grafos esparsos





Spanners

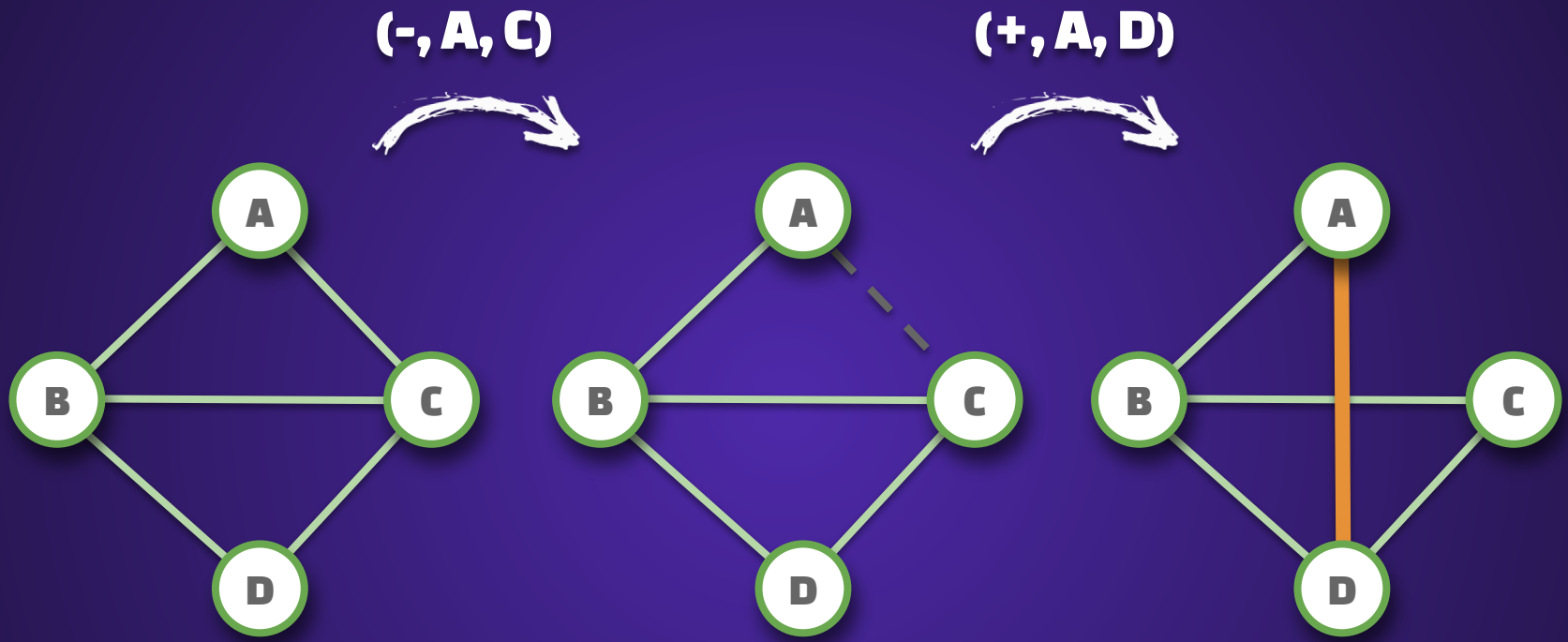
Preserva os caminhos mais curtos.

Cut sparsifiers

Preserva os cortes do grafo.

Spectral sparsifiers

Preserva propriedades da matriz laplaciana do grafo (inclui os cortes).



GRAFOS DINÂMICOS


DETECÇÃO DE BIPARTIDOS
k-CONECTIVIDADE
CONECTIVIDADE
EMPARELHAMENTO MÁXIMO
ÁRVORE GERADORA MÍNIMA

CONECTIVIDADE

EPSTEIN, 1992

Atualização: O(√n)
Consulta: O(log n)
Memória: O(m)

Amortizada

HOLM, 1998

Atualização: O(log² n),
Consulta: O(log n)
Memória: O(m)

https://dl.acm.org/citation.cfm?id=2627694

CONECTIVIDADE

Amortizada

EPSTEIN, 1992

Atualização: $O(\sqrt{n})$
Consulta: $O(\log n)$
Memória: $O(m)$

HOLM, 1998

Atualização: $O(\log^2 n)$,
Consulta: $O(\log n)$
Memória: $O(m)$

KAPRON, 2013

Atualização: $O(\log^5 n)$
Consulta: $O(\log^2 n)$
Memória: $O(n \log^3 n)$

https://dl.acm.org/citation.cfm?id=2627694

CONECTIVIDADE

Amortizada

EPSTEIN, 1992

Atualização: $O(\sqrt{n})$
Consulta: $O(\log n)$
Memória: $O(m)$

HOLM, 1998

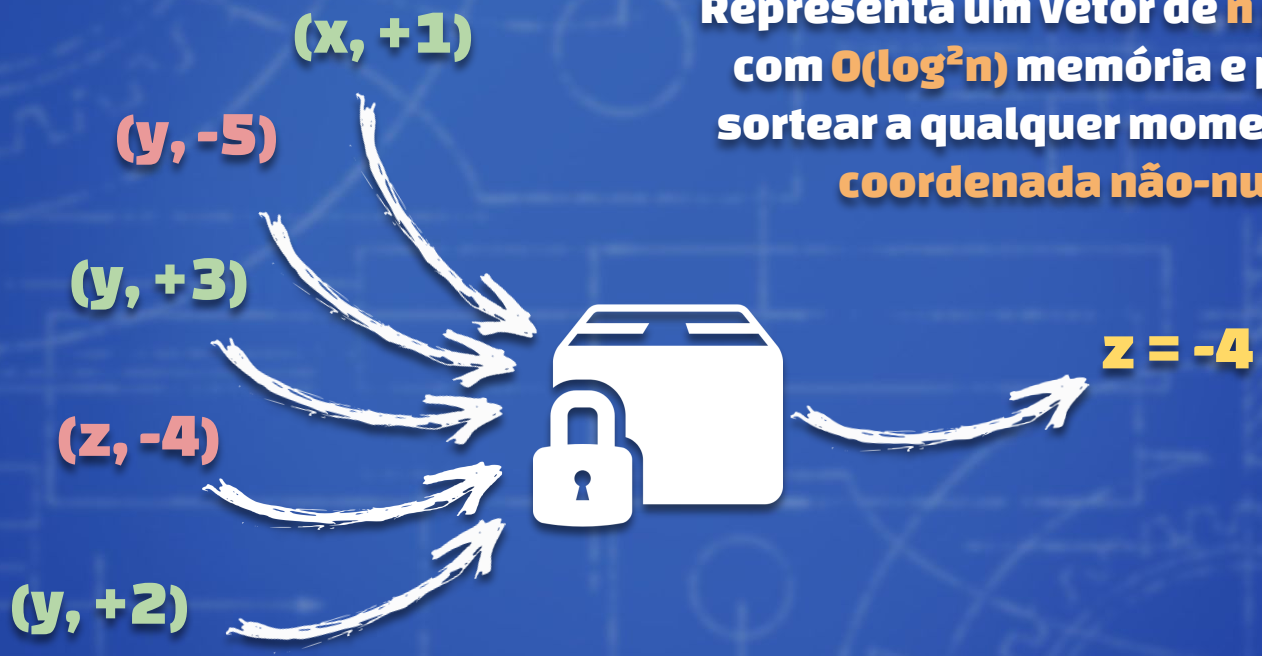
Atualização: $O(\log^2 n)$,
Consulta: $O(\log n)$
Memória: $O(m)$

AHN, 2012

Atualização: $O(\log^2 n)$
Consulta: $O(n \log^2 n)$
Memória: $O(n \log^3 n)$

https://dl.acm.org/citation.cfm?id=2627694

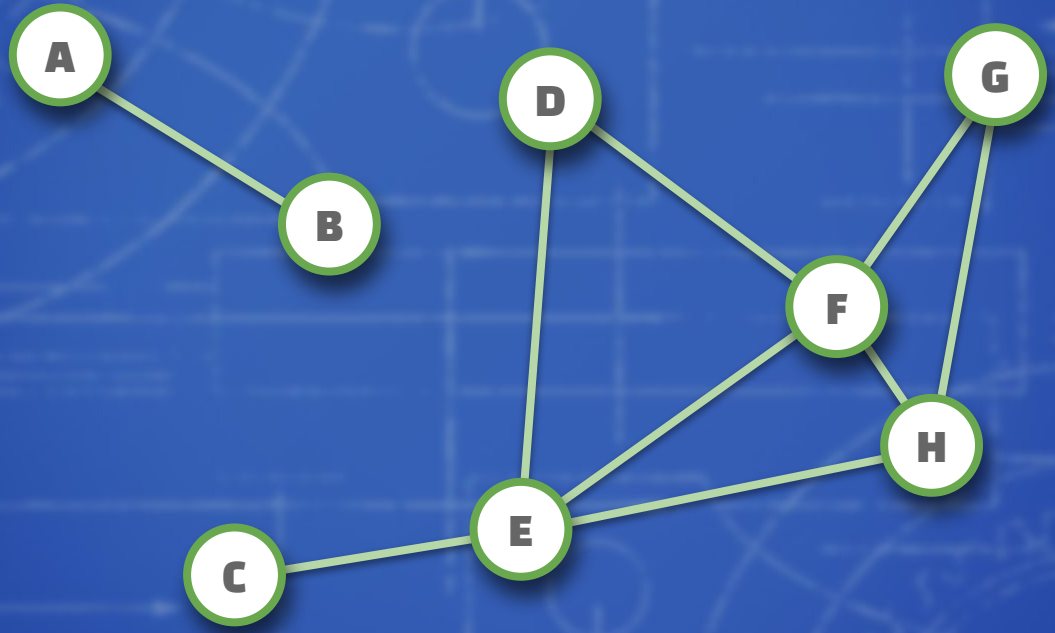
ℓ0-sampler



Representa um vetor de n posições com $O(\log^2 n)$ memória e permite sortear a qualquer momento uma coordenada não-nula.

https://dl.acm.org/citation.cfm?id=1083599

ESTE GRAFO NÃO É CONEXO

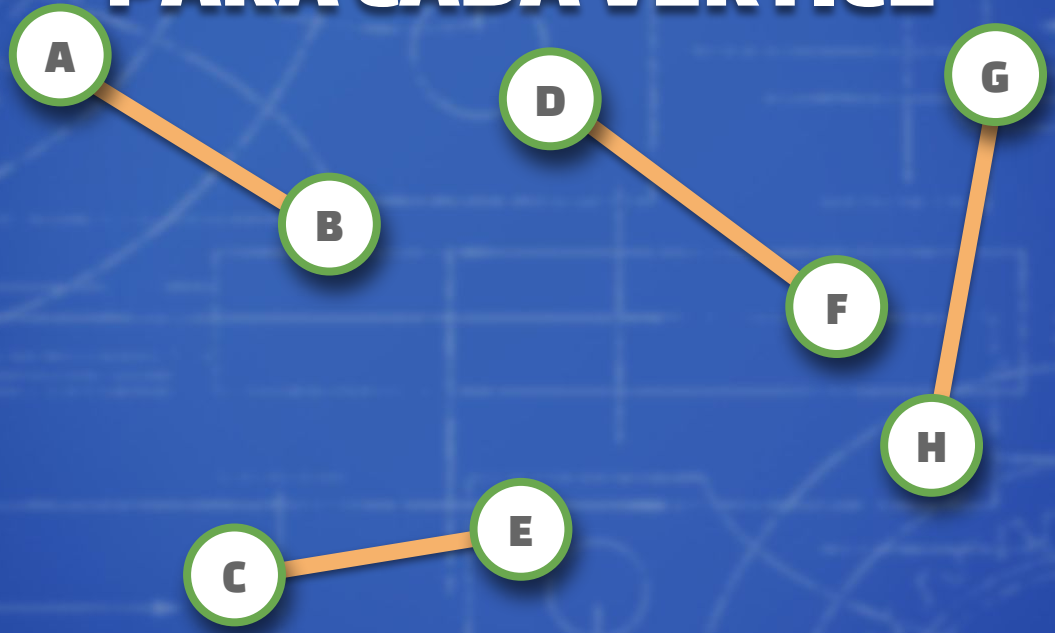


https://dl.acm.org/citation.cfm?id=2095156
https://people.cs.umass.edu/~mcgregor/papers/12-dynamic.pdf

PARTINDO DO GRAFO VAZIO

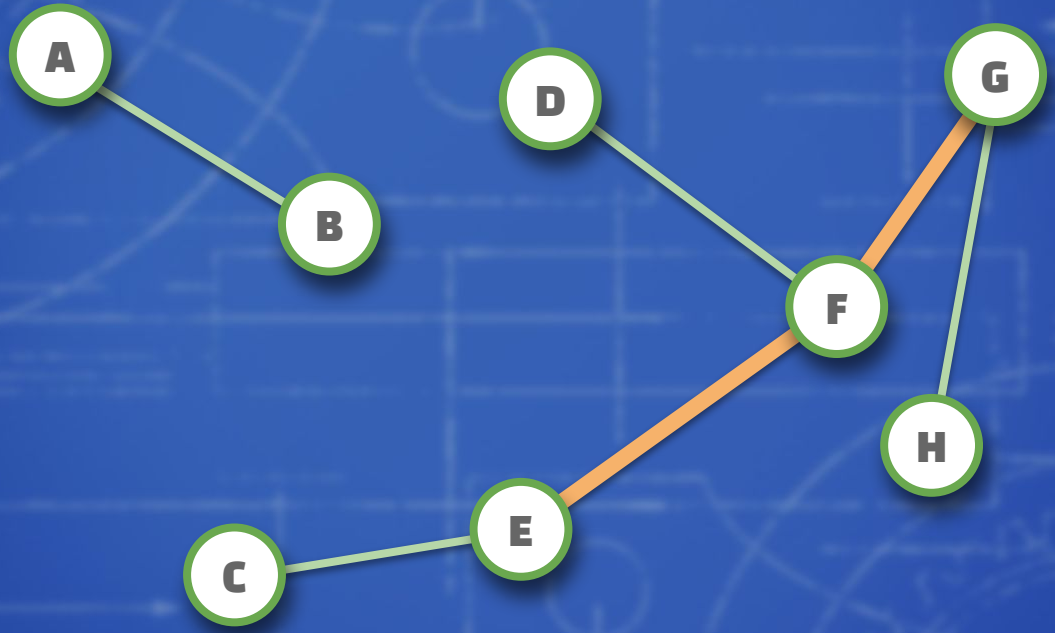
https://dl.acm.org/citation.cfm?id=2095156
https://people.cs.umass.edu/~mcgregor/papers/12-dynamic.pdf

UMA ARESTA É SORTEADA PARA CADA VÉRTICE



https://dl.acm.org/citation.cfm?id=2095156
https://people.cs.umass.edu/~mcgregor/papers/12-dynamic.pdf

MAIS UMA VEZ



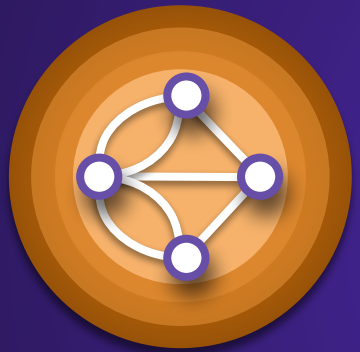
https://dl.acm.org/citation.cfm?id=2095156
https://people.cs.umass.edu/~mcgregor/papers/12-dynamic.pdf

ESTA É UMA FLORESTA GERADORA DO GRAFO

É fácil demonstrar que este algoritmo precisa de O(log n) passos no máximo.

https://dl.acm.org/citation.cfm?id=2095156
https://people.cs.umass.edu/~mcgregor/papers/12-dynamic.pdf


REPRESENTAÇÕES COMPACTAS
ESTRUTURAS PROBABILÍSTICAS
ALGORITMOS PARALELIZÁVEIS
CONSTRUÇÃO INCREMENTAL



DÚVIDAS?

Estes slides estão disponíveis em
juanlopes.net/qconsp2018