

Introdução à Inteligência Artificial – Trabalho 2

Relatório de Implementação

Alunos: Gabriel Toschi de Oliveira - Nº USP 9763039 | Marcos Vinicius Volpato - Nº USP 9364872

Introdução

Este trabalho de implementação tem como objetivo principal o estudo e teste de diferentes algoritmos e técnicas de busca em grafos, estudados durante as aulas da disciplina: busca em largura (BFS), busca em profundidade (DFS), busca best-first, busca A e busca A*. Como base para os testes, todos os algoritmos foram executados em cima de grafos-knn, gerados aleatoriamente para cada teste. Foram coletados como resultados dos testes o tempo de execução dos algoritmos em cada cenário de teste, assim como uma representação gráfica dos caminhos feitos por ele.

As próximas seções apresentarão detalhes da implementação feita durante o trabalho, os resultados coletados durante os testes e uma discussão sobre estes dados.

Detalhes da implementação

De acordo com o recomendado pela descrição do trabalho, toda a implementação dos algoritmos e testes foi feita na linguagem Python 3. Optou-se por usar a menor quantidade possível de bibliotecas externas e priorizar a codificação de todos os algoritmos manualmente — códigos externos foram usados apenas para a geração das imagens dos grafos e dos caminhos percorridos. As instruções para a execução dos testes em ambientes Linux está disponível no arquivo `README.md`. [O código-fonte também está disponível no GitHub.](#)

A implementação foi realizada em quatro passos principais: (1) criação de TADs para grafos e grafos-knn (arquivo `graph.py`); (2) implementação dos algoritmos de busca; (3) geração das imagens dos grafos e caminhos; e (4) automatização dos testes realizados. Com os grafos já bem definidos em código, o passo seguinte foi realizar a implementação dos diversos algoritmos de busca a serem testados. Todos foram feitos em módulos separados e utilizando a mesma assinatura de função, permitindo sua agilidade na automação posterior dos testes.

Enquanto alguns dos algoritmos de busca são conhecidos por especificações bem definidas, as buscas A e A* recebem seus nomes de acordo com a heurística escolhida para compor suas verificações. Seguindo orientações do professor e do estagiário PAE da disciplina sobre quais heurísticas seriam interessantes para serem escolhidas, optou-se por usar a distância euclidiana, classicamente conhecida como uma heurística admissível, para a busca A*. Para os testes da busca A, foram feitas duas versões, uma utilizando a distância Manhattan e outra utilizando o décupo (10 vezes) da distância euclidiana. Ambas as versões foram separadamente testadas e analisadas.

Desta forma, têm-se a lista dos algoritmos codificados para testes neste trabalho:

- Busca em largura (BFS) — arquivo `BFS.py`;
- Busca em profundidade (DFS) — arquivo `DFS.py`;
- Busca best-first — arquivo `bestfirst.py`;
- Busca A (usando a distância Manhattan) — arquivo `amanhattan.py`;
- Busca A (usando 10x a distância euclidiana) — arquivo `aeuclidian.py`;
- Busca A* (usando a distância euclidiana) — arquivo `astar.py`.

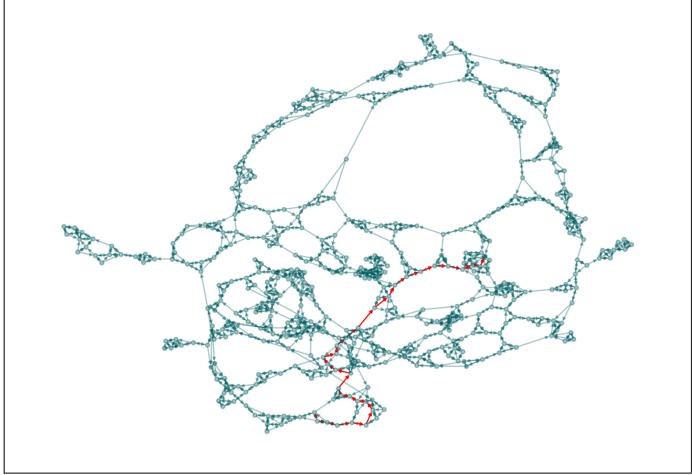
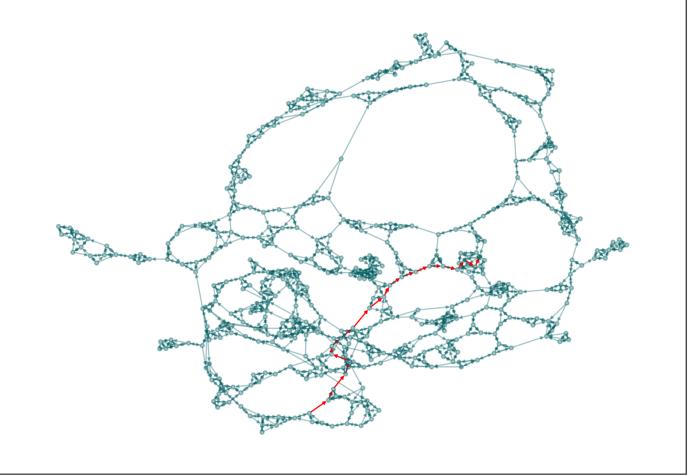
Por fim, foi criado um algoritmo capaz de gerar um grafo-knn seguindo os valores **v** e **k** indicados, decidir aleatoriamente um vértice de início e fim para as buscas, fazer os testes necessários e coletar as informações de tempo de execução e caminho percorridos. Este algoritmo está disponível no arquivo `tests.py`. A fim de minimizar possíveis diferenças nos tempos registrados, o mesmo grafo-knn e os mesmos vértices de início e fim foram usados para o teste de todos os algoritmos e cada algoritmo foi executado 5 vezes e a duração registrada foi a média aritmética dos resultados em cada uma das execuções.

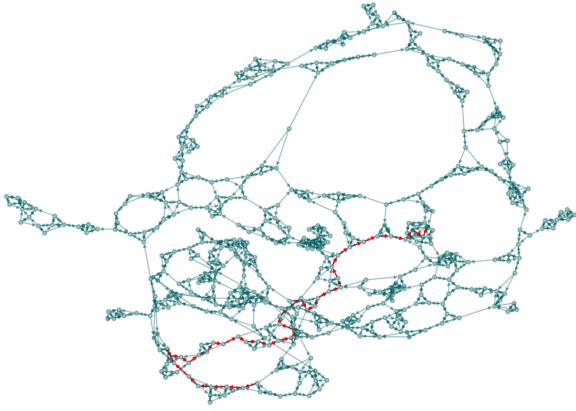
Resultados coletados

Para realizar os testes, foram escolhidos conjuntos de valores para serem atribuídos a **v** e **k** para produzir diferentes cenários de teste — totalizando 9 cenários de teste. Os dados coletados foram o tempo médio de execução da busca (em milissegundos), o tamanho (em vértices) do caminho encontrado e a representação visual do caminho (em vermelho). As imagens e os dados coletados estão presentes em maior resolução e detalhamento na pasta `results`. Todos os algoritmos foram testados em todos os cenários:

- Número de vértices (**v**): 500, 1250 e 2500 vértices
- Número de arestas (**k**): 3, 5 e 7 arestas por vértice

v = 500, k = 3

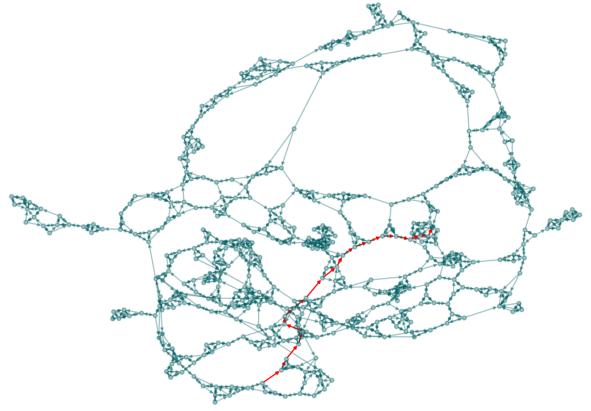
DFS (busca em profundidade)		BFS (busca em largura)	
0,03924 ms	31 vértices	0,10040 ms	20 vértices
			
Busca best-first		Busca A* (distância euclidiana)	
0,63419 ms	42 vértices	0,43998 ms	20 vértices



Busca A (distância Manhattan)

0,25554 ms

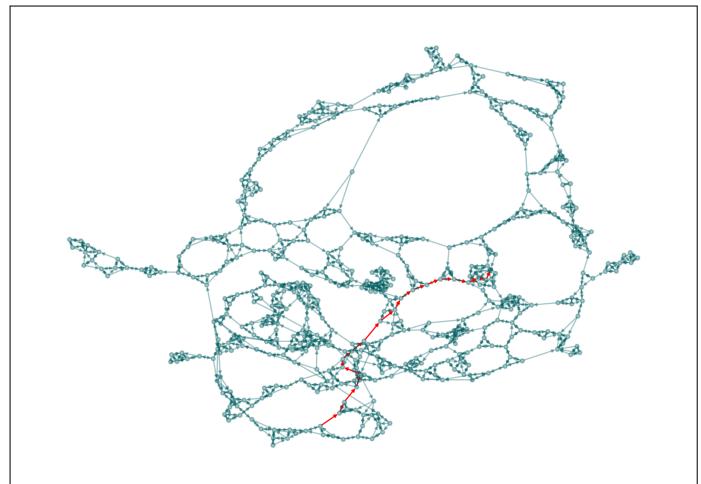
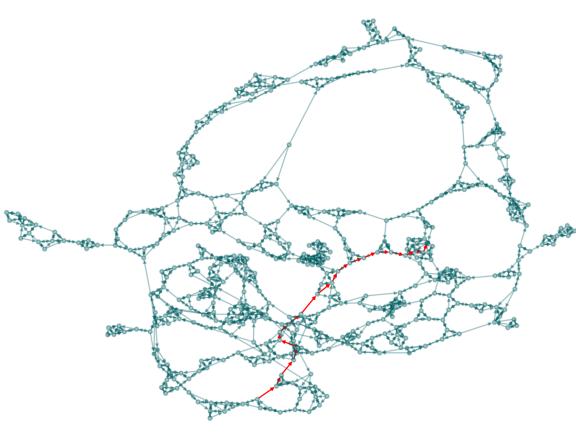
20 vértices



Busca A (10x distância euclidiana)

0,19932 ms

20 vértices



$v = 500, k = 5$

DFS (busca em profundidade)

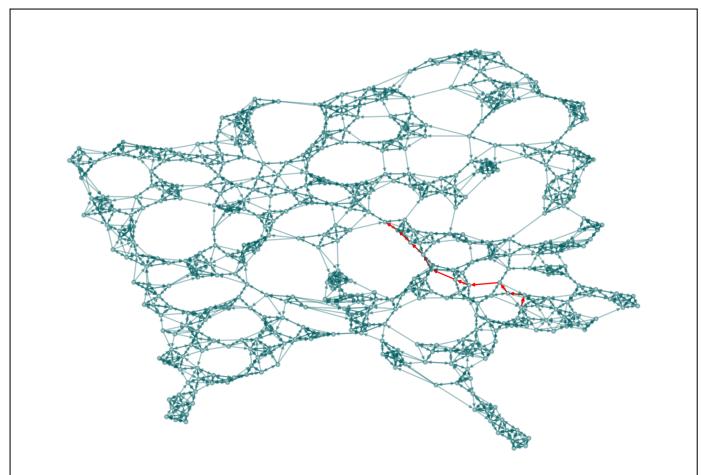
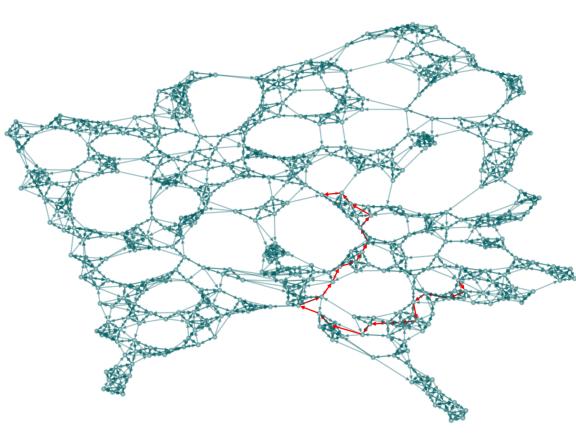
0,00024 ms

24 vértices

BFS (busca em largura)

0,08626 ms

11 vértices



Busca best-first

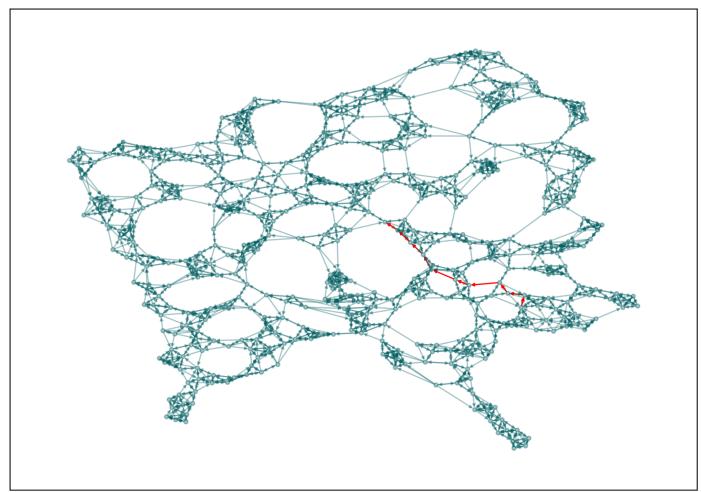
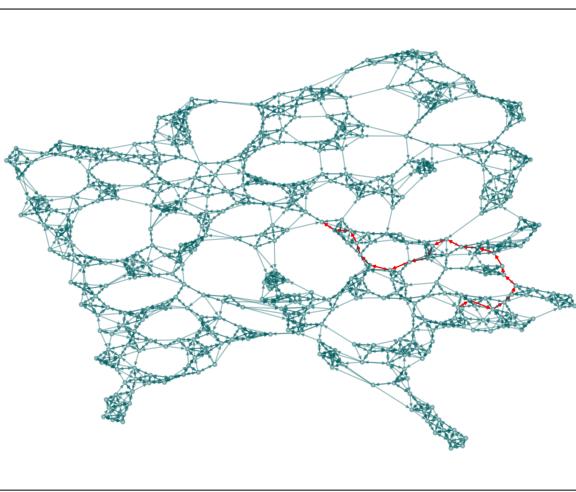
0,00087 ms

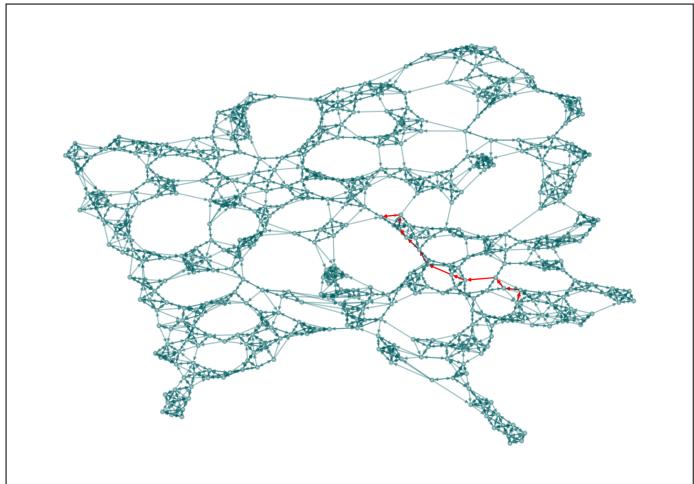
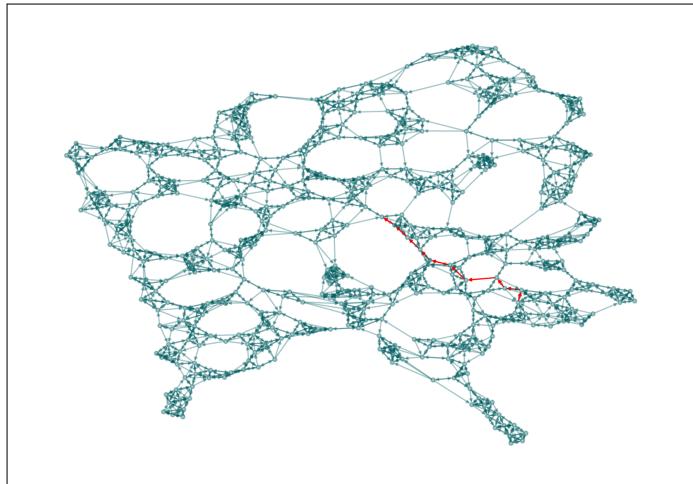
22 vértices

Busca A* (distância euclidiana)

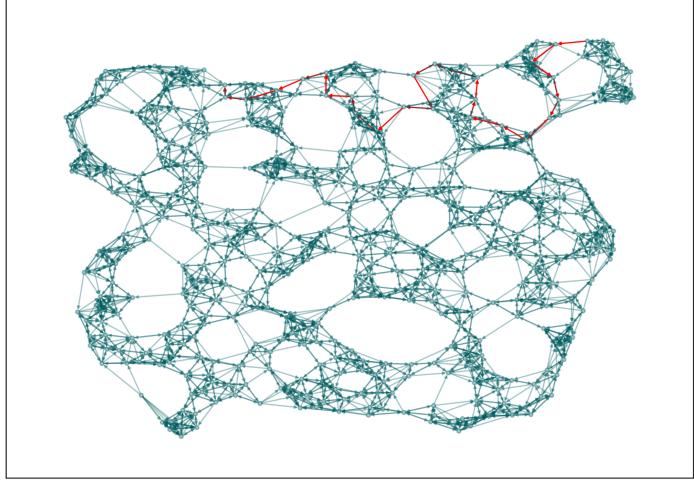
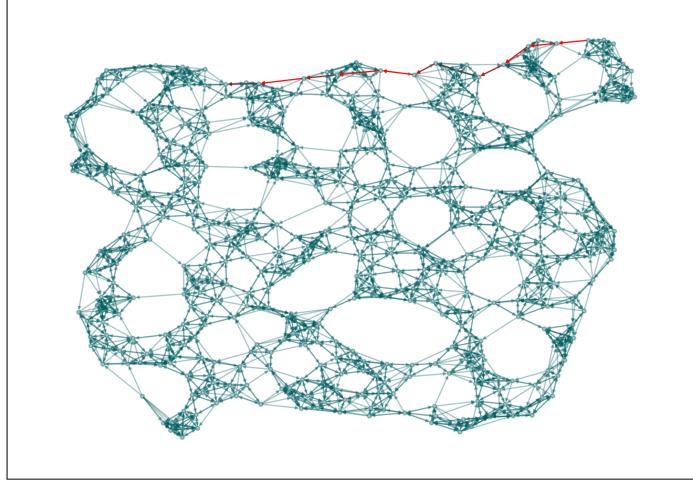
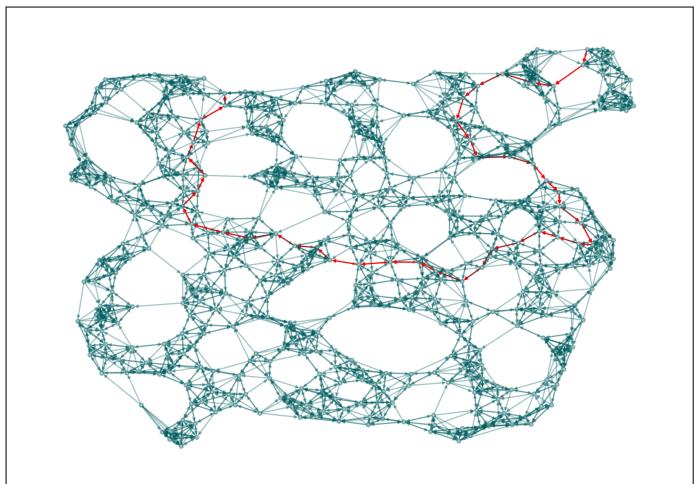
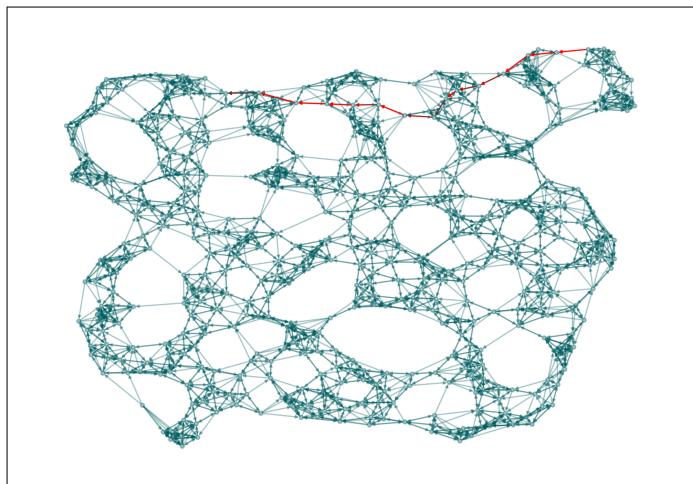
0,00021 ms

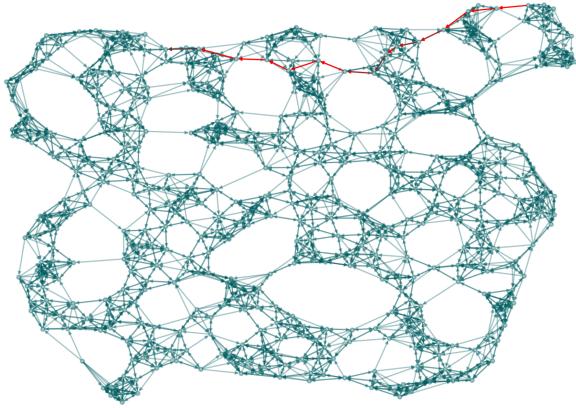
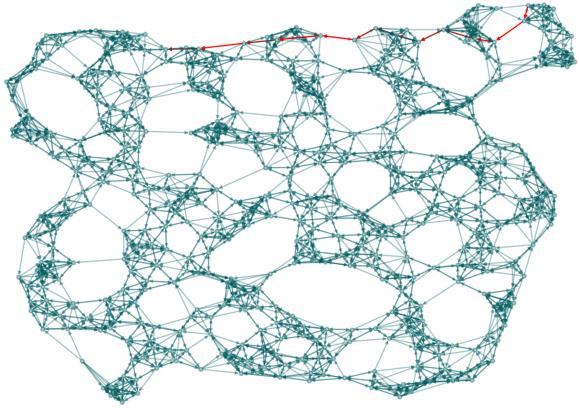
11 vértices



Busca A (distância Manhattan)		Busca A (10x distância euclidiana)	
0,00015 ms	12 vértices	0,00015 ms	11 vértices
			

$v = 500, k = 7$

DFS (busca em profundidade)		BFS (busca em largura)	
0,36535 ms	26 vértices	0,12422 ms	13 vértices
			
0,11824 ms	43 vértices	0,44904 ms	15 vértices
Busca best-first		Busca A* (distância euclidiana)	
0,11824 ms	43 vértices	0,44904 ms	15 vértices
			
Busca A (distância Manhattan)		Busca A (10x distância euclidiana)	
0,41547 ms	14 vértices	0,29220 ms	15 vértices



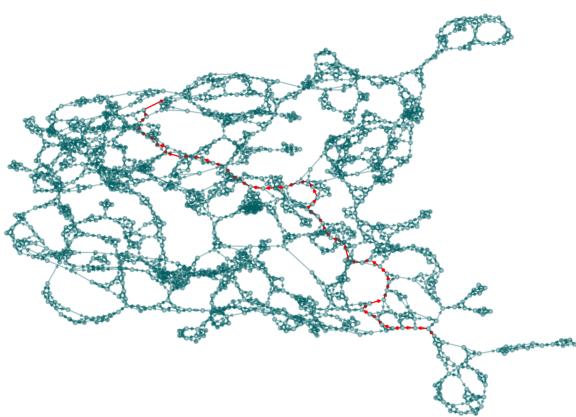
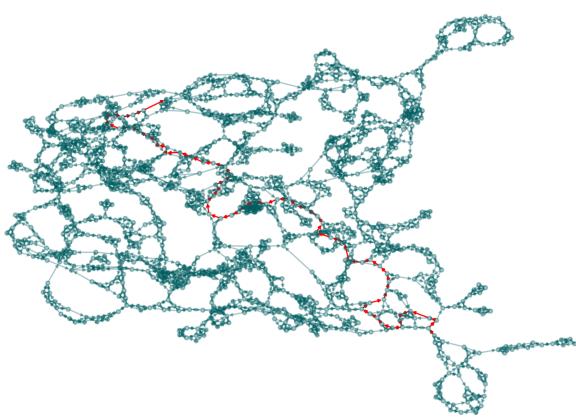
$v = 1250, k = 3$

DFS (busca em profundidade)

0,11563 ms | 64 vértices

BFS (busca em largura)

0,16160 ms | 50 vértices

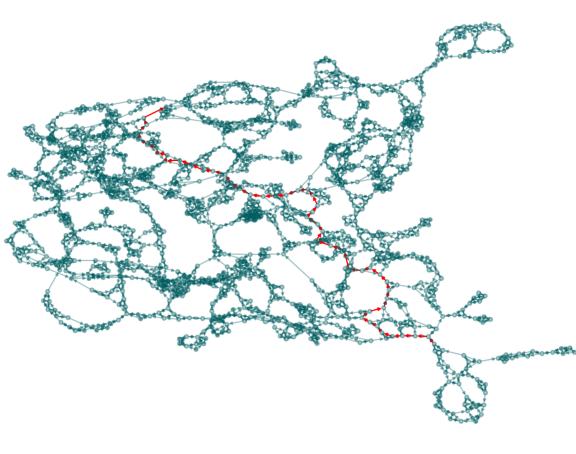
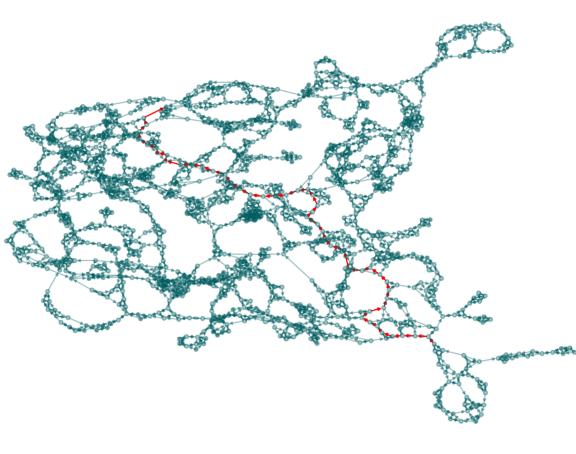


Busca best-first

1,15675 ms | 50 vértices

Busca A* (distância euclidiana)

1,99203 ms | 50 vértices

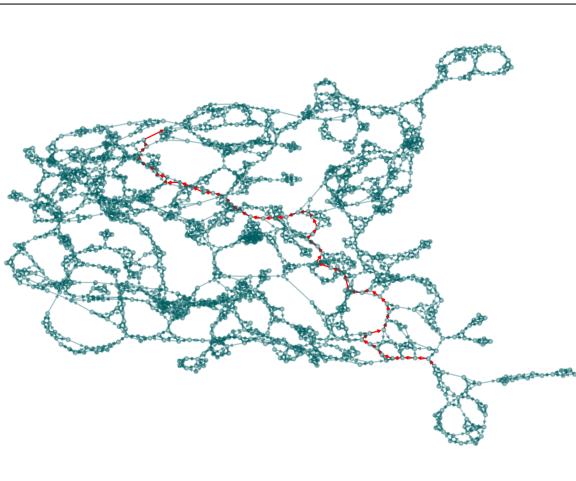
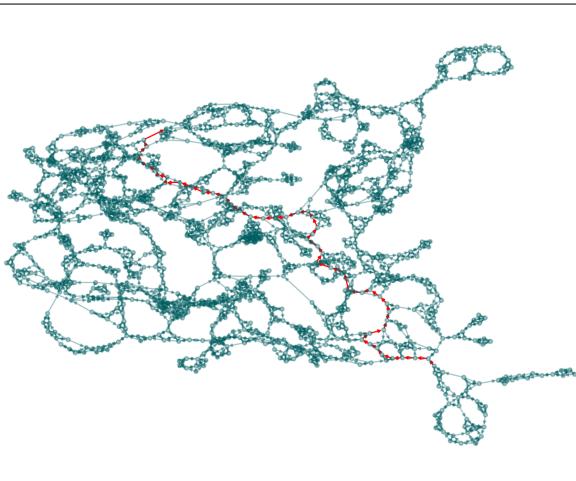


Busca A (distância Manhattan)

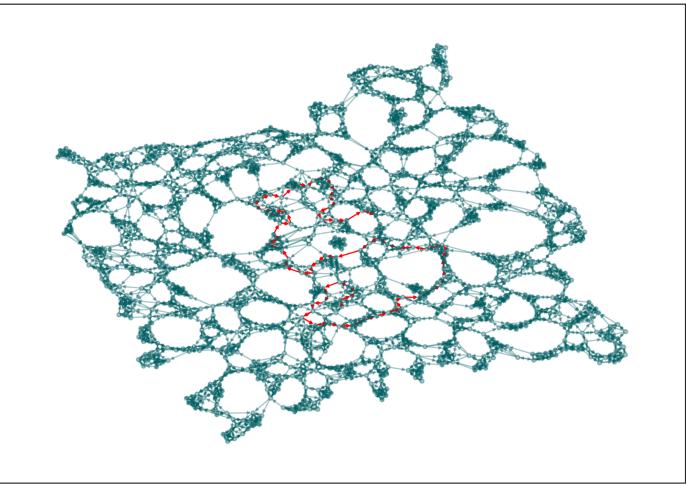
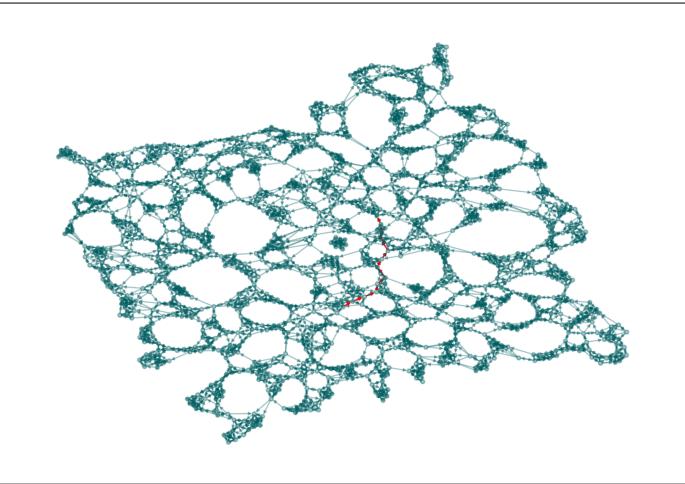
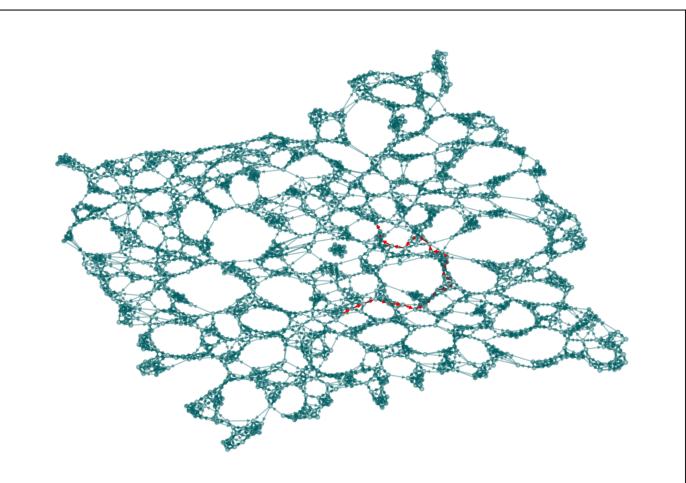
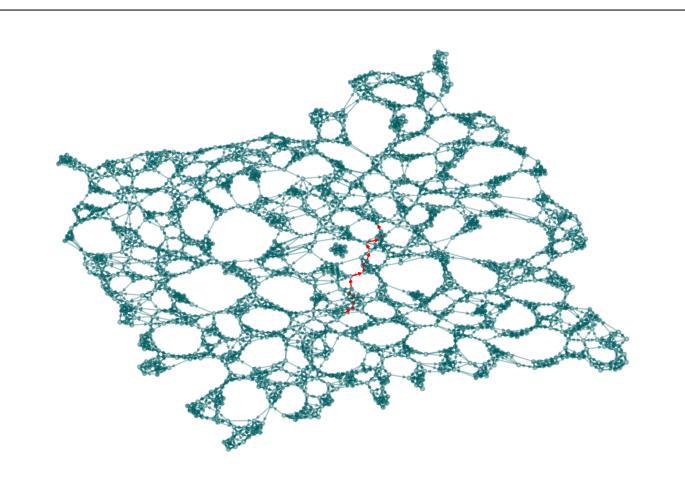
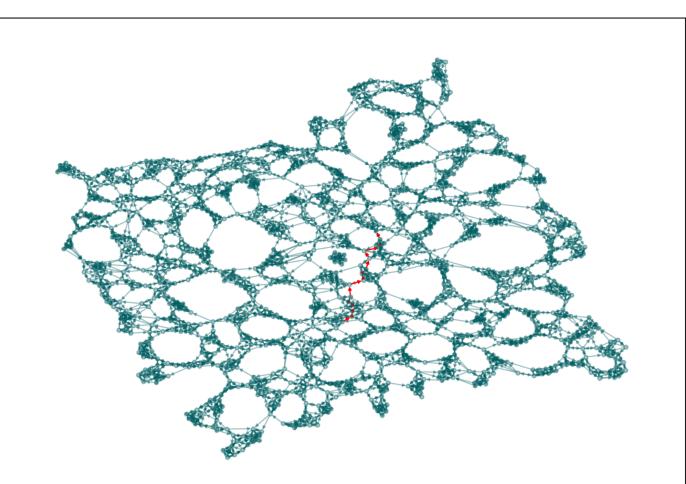
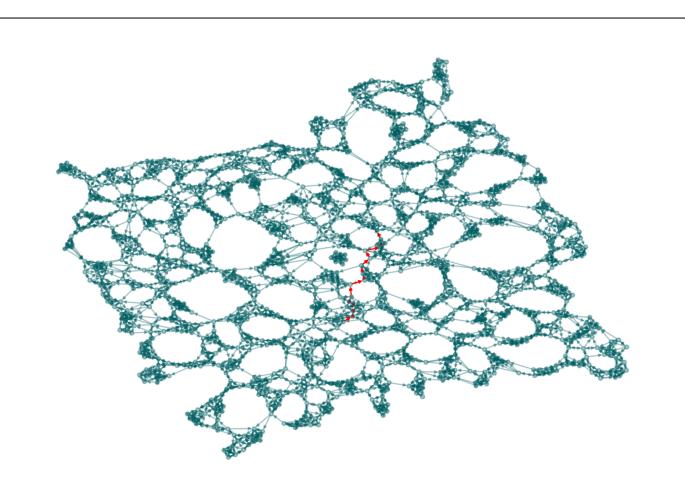
2,33297 ms | 50 vértices

Busca A (10x distância euclidiana)

1,19376 ms | 50 vértices

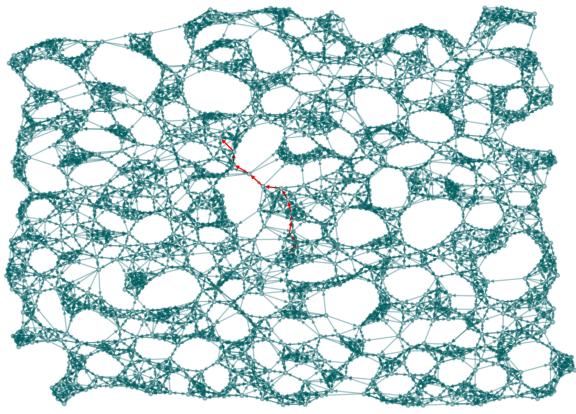
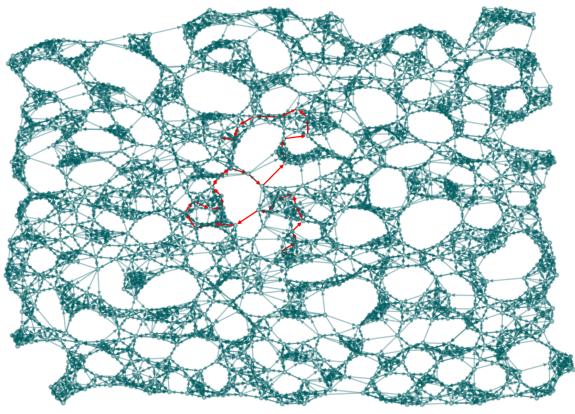


$v = 1250, k = 5$

DFS (busca em profundidade)		BFS (busca em largura)	
1,04103 ms	74 vértices	0,27194 ms	13 vértices
			
Busca best-first		Busca A* (distância euclidiana)	
1,25980 ms	26 vértices	0,41875 ms	15 vértices
			
Busca A (distância Manhattan)		Busca A (10x distância euclidiana)	
0,20795 ms	15 vértices	0,20704 ms	15 vértices
			

$v = 1250, k = 7$

DFS (busca em profundidade)		BFS (busca em largura)	
0,04110 ms	34 vértices	0,37946 ms	13 vértices



Busca best-first

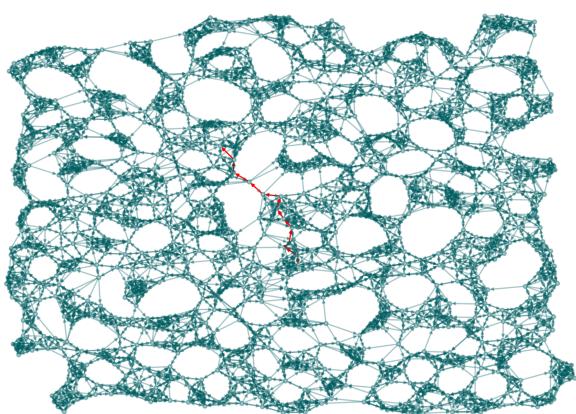
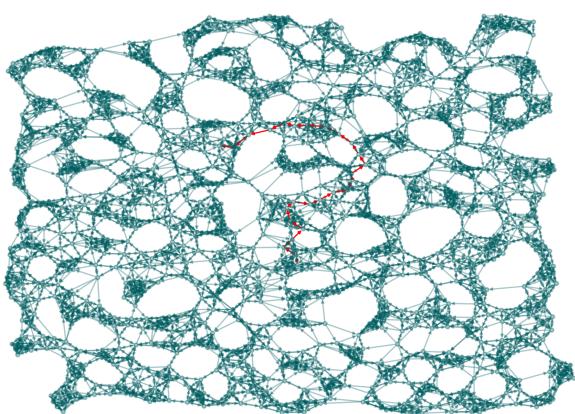
4,65307 ms

26 vértices

Busca A* (distância euclidiana)

0,40259 ms

13 vértices



Busca A (distância Manhattan)

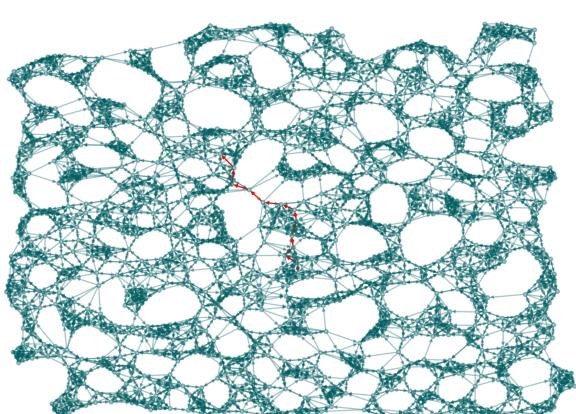
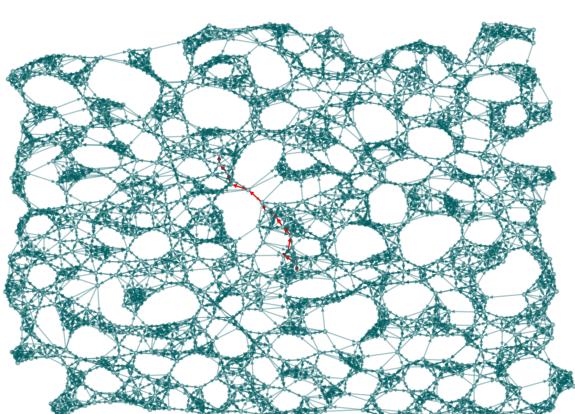
0,41050 ms

15 vértices

Busca A (10x distância euclidiana)

0,24700 ms

13 vértices



$v = 2500, k = 3$

DFS (busca em profundidade)

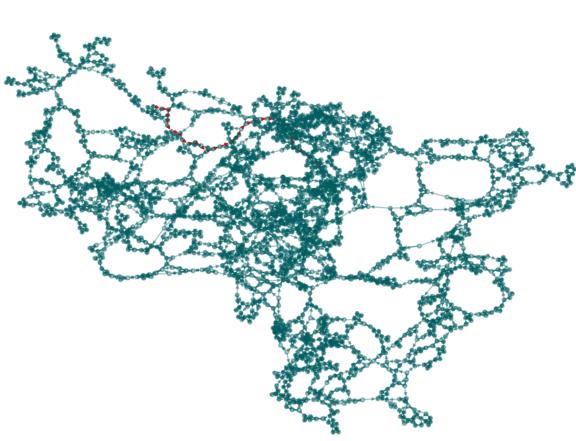
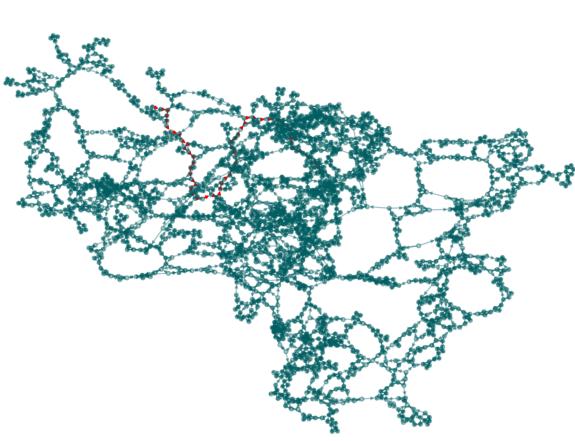
0,20208 ms

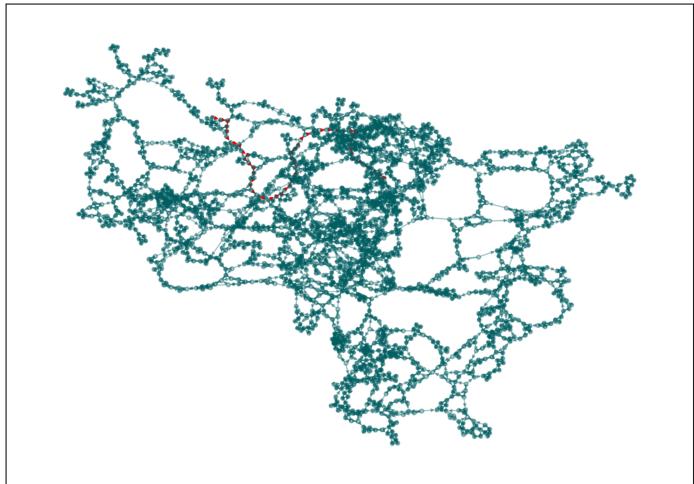
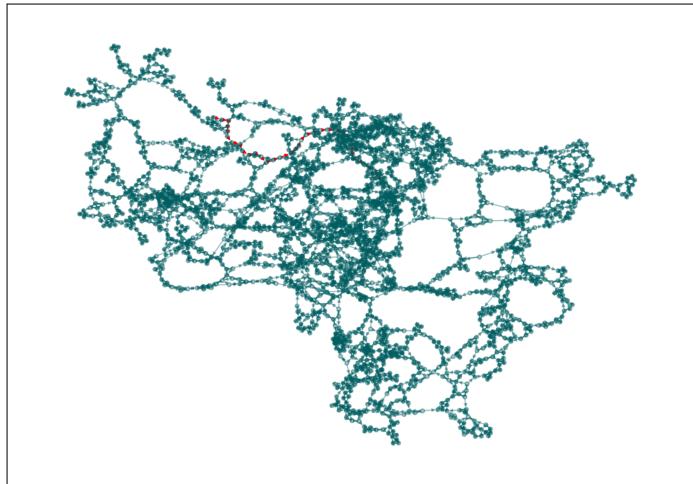
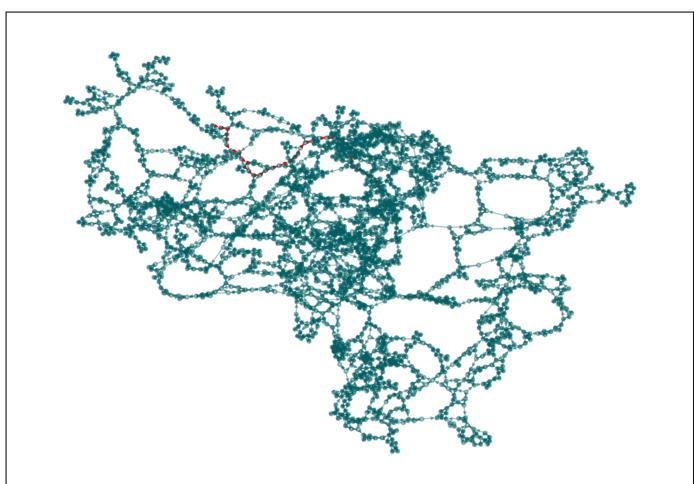
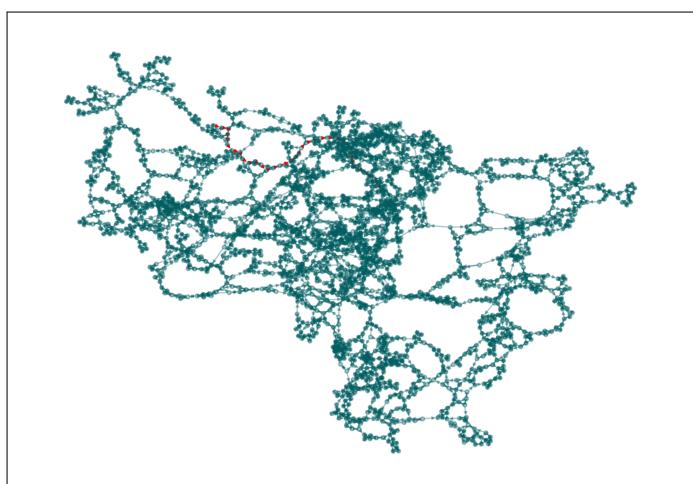
53 vértices

BFS (busca em largura)

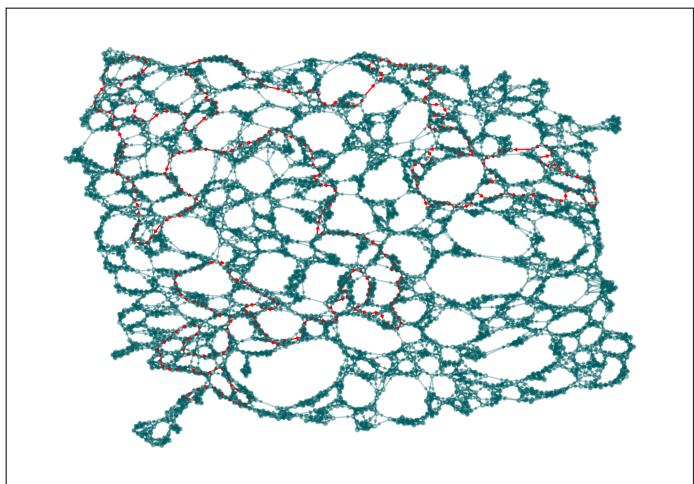
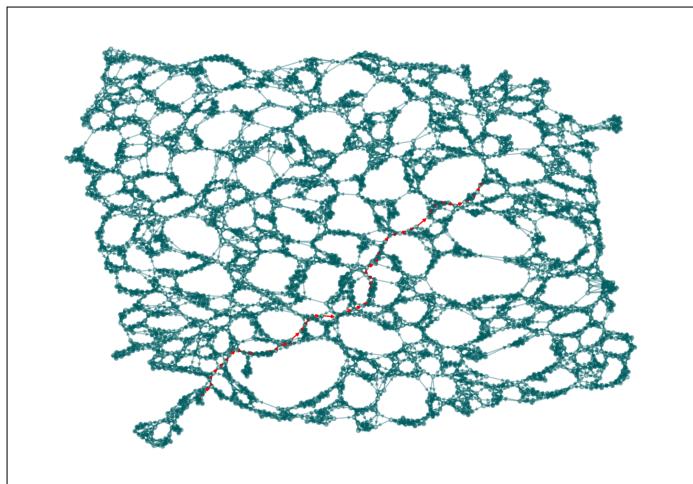
0,16765 ms

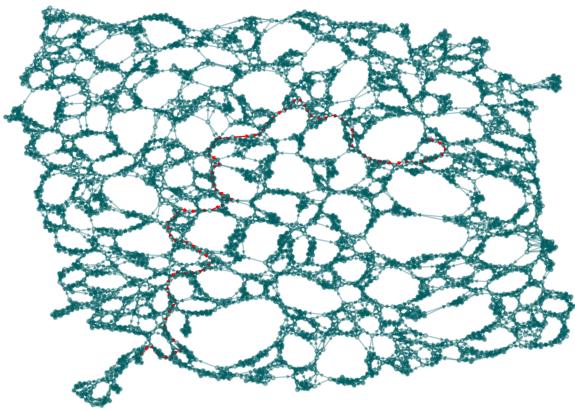
35 vértices



Busca best-first		Busca A* (distância euclidiana)	
1,78976 ms	50 vértices	1,25460 ms	35 vértices
			
Busca A (distância Manhattan)		Busca A (10x distância euclidiana)	
1,16353 ms	37 vértices	0,49863 ms	35 vértices
			

$v = 2500, k = 5$

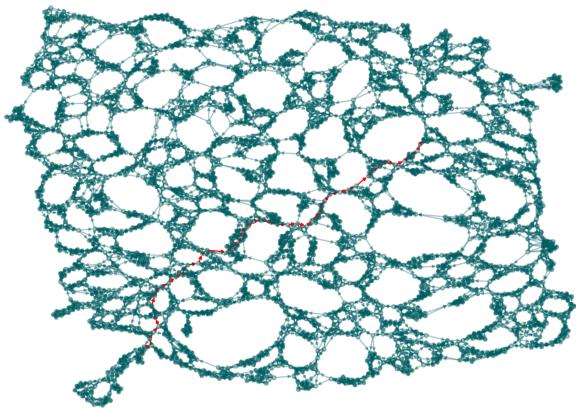
DFS (busca em profundidade)		BFS (busca em largura)	
1,23839 ms	377 vértices	1,27205 ms	50 vértices
			
Busca best-first		Busca A* (distância euclidiana)	
10,77289 ms	102 vértices	6,23621 ms	52 vértices



Busca A (distância Manhattan)

0,95705 ms

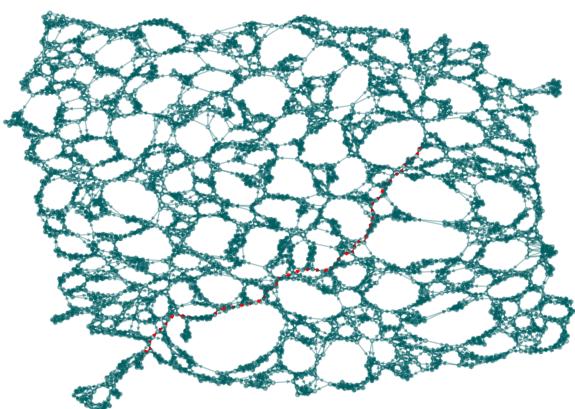
52 vértices



Busca A (10x distância euclidiana)

0,77395 ms

55 vértices



$v = 2500, k = 7$

DFS (busca em profundidade)

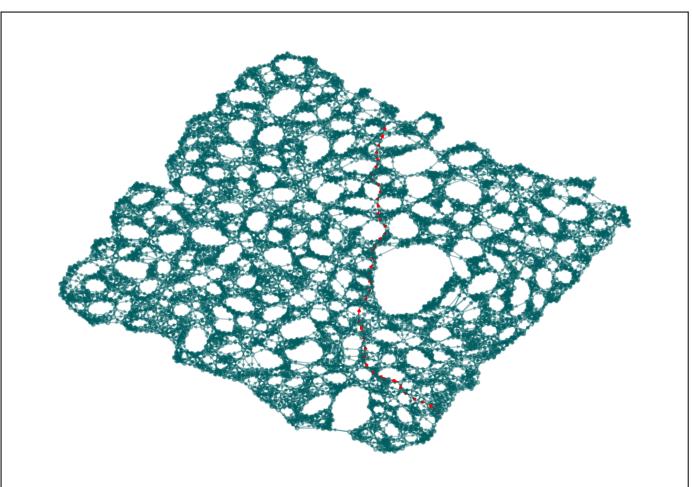
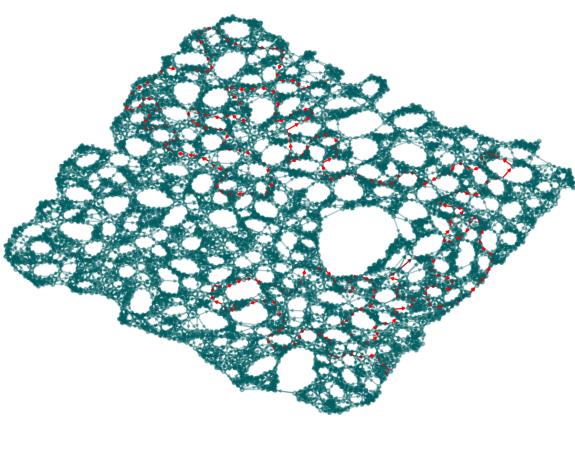
1,34363 ms

290 vértices

BFS (busca em largura)

1,89037 ms

43 vértices



Busca best-first

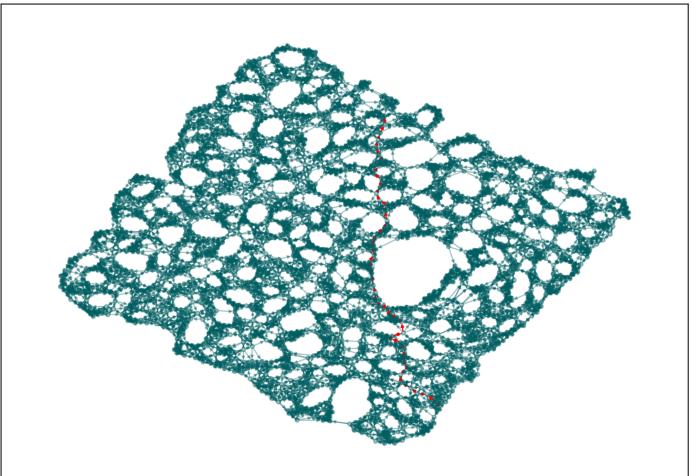
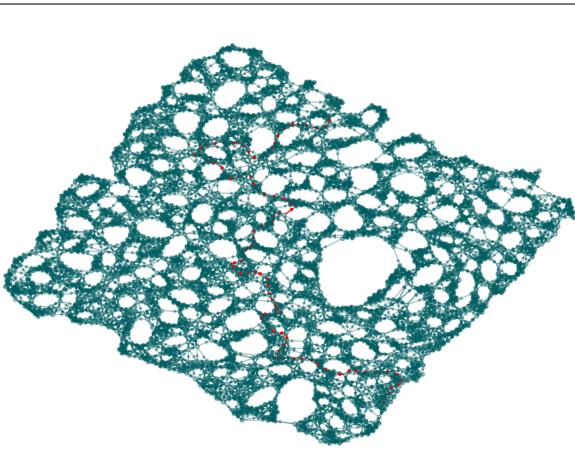
8,74752 ms

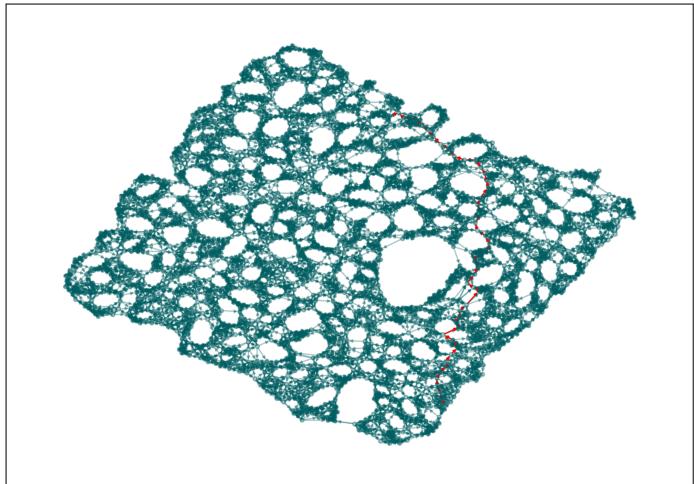
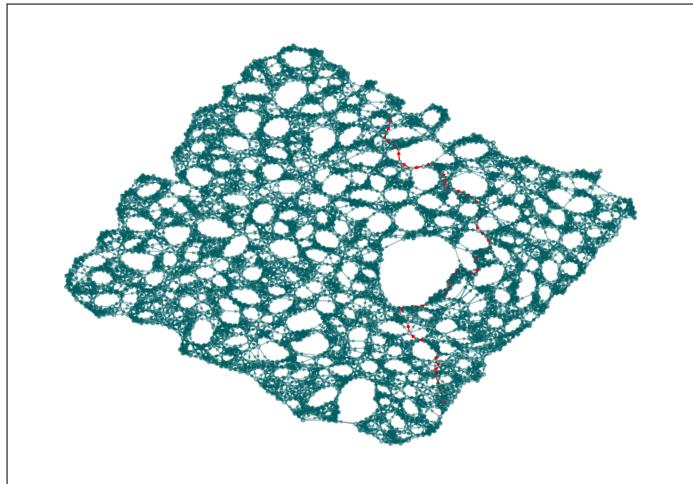
94 vértices

Busca A* (distância euclidiana)

4,68597 ms

47 vértices



Busca A (distância Manhattan)		Busca A (10x distância euclidiana)	
1,26567 ms	49 vértices	1,07546 ms	57 vértices
			

Resumo dos resultados

	DFS	BFS	Best-First	A*	A (Manhattan)	A (10*Euclidiana)
v = 500, k = 3	0,03924 ms / 31 v	0,10040 ms / 20 v	0,63419 ms / 42 v	0,43998 ms / 20 v	0,25554 ms / 20 v	0,19932 ms / 20 v
v = 500, k = 5	0,00024 ms / 24 v	0,08626 ms / 11 v	0,00087 ms / 22 v	0,00021 ms / 11 v	0,00015 ms / 12 v	0,00015 ms / 11 v
v = 500, k = 7	0,36535 ms / 26 v	0,12422 ms / 13 v	0,11824 ms / 43 v	0,44904 ms / 15 v	0,41547 ms / 14 v	0,29220 ms / 15 v
v = 1250, k = 3	0,11563 ms / 64 v	0,16160 ms / 50 v	1,15675 ms / 50 v	1,99203 ms / 50 v	2,33297 ms / 50 v	1,19376 ms / 50 v
v = 1250, k = 5	1,04103 ms / 74 v	0,27194 ms / 13 v	1,25980 ms / 26 v	0,41875 ms / 15 v	0,20795 ms / 15 v	0,20704 ms / 15 v
v = 1250, k = 7	0,04110 ms / 34 v	0,37946 ms / 13 v	4,65307 ms / 26 v	0,40259 ms / 13 v	0,41050 ms / 15 v	0,24700 ms / 13 v
v = 2500, k = 3	0,20208 ms / 53 v	0,16765 ms / 35 v	1,78976 ms / 50 v	1,25460 ms / 35 v	1,16353 ms / 37 v	0,49863 ms / 35 v
v = 2500, k = 5	1,23839 ms / 377v	1,27205 ms / 50 v	10,77289 ms / 102v	6,23621 ms / 52 v	0,95705 ms / 52 v	0,77395 ms / 55 v
v = 2500, k = 7	1,34363 ms / 290v	1,89037 ms / 43 v	8,74752 ms / 94 v	4,68597 ms / 47 v	1,26567 ms / 49 v	1,07546 ms / 57 v

Discussões sobre os resultados

A partir dos resultados coletados, é possível perceber alguns padrões de execução entre os algoritmos dentro de alguns cenários. A DFS, por mais que represente alguns dos melhores resultados em tempo de execução, ela consegue apresentar, nos cenários com $v = 1250$ ou 2500 , os maiores caminhos encontrados, as vezes até três vezes maior que os menores caminhos (como em $v = 2500, k = 5$, no qual o caminho da DFS passa por 377 vértices contra 50 vértices da BFS). Um indício deste comportamento que pode ser relacionado é a própria natureza da busca em profundidade, que não abarca níveis completos do grafo antes de passar pro próximo.

Quanto ao tempo de execução, os algoritmos DFS e BFS e as duas versões da Busca A apresentaram os melhores tempos de busca em cenários diferentes. Enquanto para um número menor de vértices, as duas primeiras técnicas apresentaram tempos melhores, a Busca A mostrou se melhor para os cenários com maior número de vértices. Entretanto, muitas vezes a diferença não é tão grande ou significativa entre estes algoritmos. Por sua vez, a busca Best-First apresentou os piores tempos de execução em todos os cenários, exceto pela situação $v = 500, k = 7$ — no pior caso ($v = 2500, k = 5$), chega a ser cerca de 15 vezes maior que o melhor desempenho, feito pela Busca A (10*Euclidiana). Uma possível causa seria o custo em manter uma lista de prioridades de nós sem o uso de uma heurística, como é feito na Busca A e A*.

Um dos cenários a serem observados é a de uma possível diferença entre as heurísticas utilizadas pelas Buscas A* e A, questão de dúvida na implementação do trabalho. Enquanto ambas tiveram tamanhos de caminho bem próximos (inclusive, formando exatamente os mesmos caminhos em algumas situações), a Busca A* sempre teve um desempenho pior do que alguma das heurísticas usadas para a Busca A. Entre as duas heurísticas usadas para a busca A, o déncuplo da distância euclidiana apresentou um desempenho melhor ou igual do que a distância Manhattan, tanto em tempo de execução quanto em tamanho do caminho.

Este resultado favorável à busca A (com o déncuplo da distância euclidiana) pode também ser resultado da natureza do grafo-knn, cujas arestas estão sempre conectando os vértices mais próximos. Aumentar o peso da distância na heurística (multiplicando por 10) pode ter acarretado uma maior tendência a procurar vértices mais perto do atual do que outras heurísticas.

Inclusive, é possível observar que, exceto pelo tempo de execução nos últimos dois cenários ($v = 2500, k = 5$ e 7), o desempenho da distância euclidiana padrão e da distância Manhattan são muito parecidos. Isto pode ser visto como um indicativo forte de que a distância Manhattan seja tão otimista quanto a distância euclidiana, como indicado nas discussões entre o grupo e o estagiário PAE durante a etapa de desenvolvimento.