Nguyễn Quang Trường – 19CLC3
Student ID: 19127608

REPORT DOCUMENT

LAB 01

SEARCH STRATEGIES

I. Overview:

In this individual project, I focus on Artificial Intelligence search algorithms.

Which to be run on 3 different graphs implemented in an adjacency matrix.

Then, collect the results and write down my own comments.

So, these bellow algorithms that I have done:

- Breadth-first search. **(BFS)**
- Tree-search depth-first search. **(DFS)**
- Uninform-cost search. **(UCS)**
- Iterative deepening search. **(IDS)**

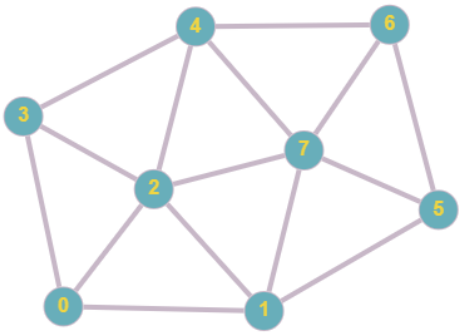
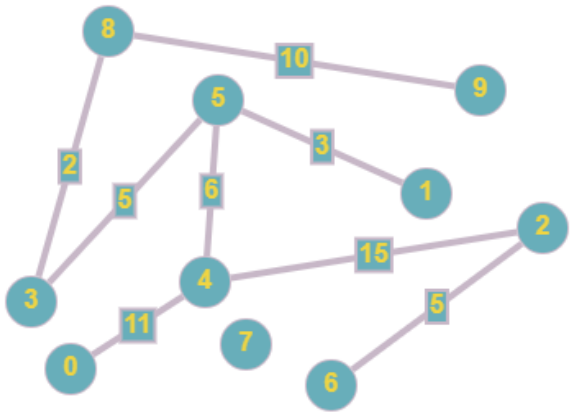
Though, what a shame that I could not handle the bellow algorithm, yet.

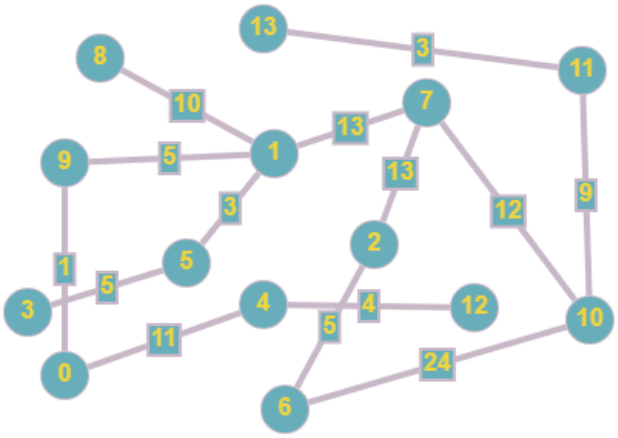
I am trying to implement this algorithm:

- Hill-climbing. **(HC)**

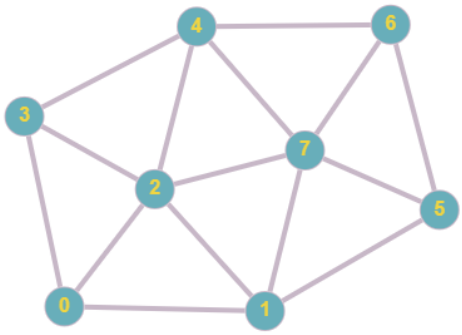
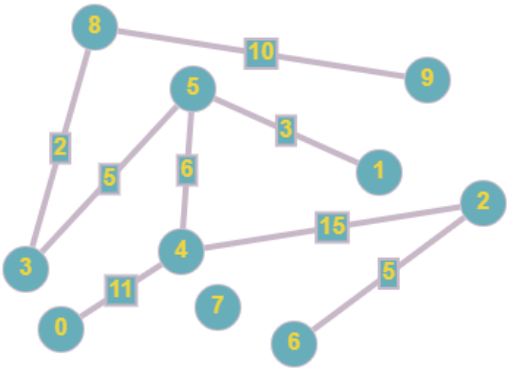
II. Execution result:

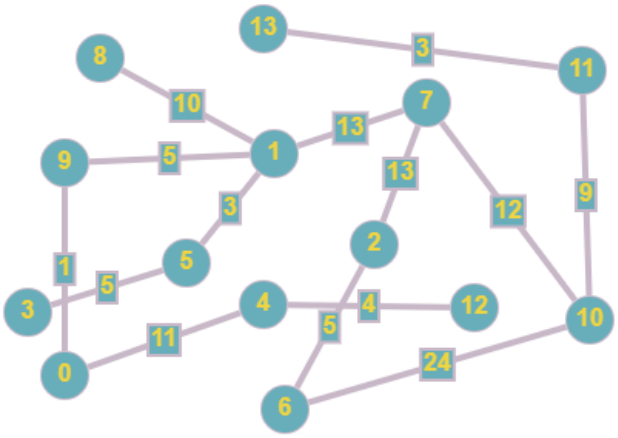
1. Breadth-first search:

Graph	Input file *input.txt*	Output file *output.txt*
	8 4 5 0 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 3 2 0 0 1 1 2 0	4 2 3 6 4 6 5
	10 5 7 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 15 0 5 0 0 0 0 0 0 0 0 5 0 0 2 0 11 0 15 0 0 6 0 0 0 0 0 3 0 5 6 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 10 0 0 0 0 0 0 0 0 10 0 12 11 7 1 5 2 3 2 12 24	5 1 3 4 8 0 2 9 6 No path exists.

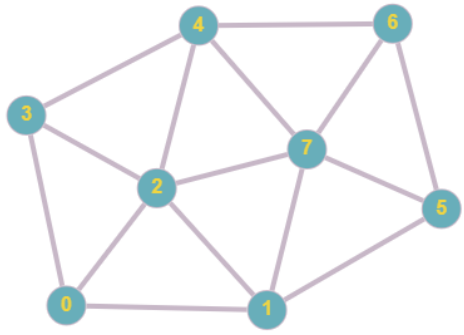
	14 13 12 0 0 0 0 0 11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 0 13 10 5 0 0 0 0 0 0 0 0 0 0 5 13 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 4 0 0 3 0 5 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 24 0 0 0 0 13 13 0 0 0 0 0 0 0 12 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 24 12 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 12 11 5 7 1 5 8 9 19 8 6 12 4 1	13 11 10 6 7 2 1 5 8 9 3 0 4 13 11 10 7 1 9 0 4 12

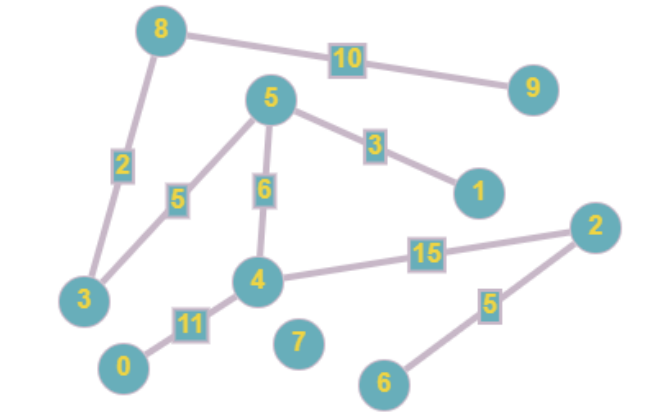
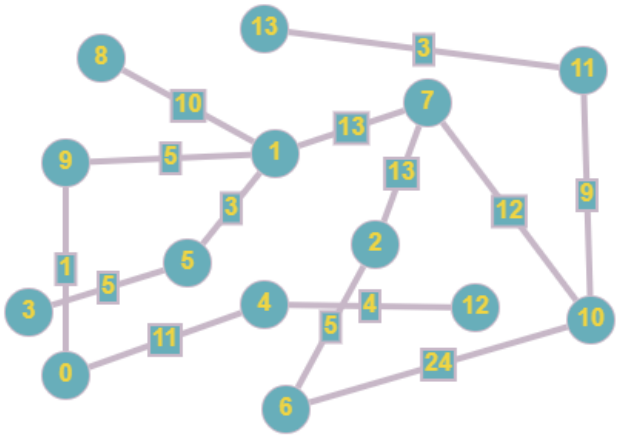
2. Tree-search depth-first search:

Graph	Input file *input.txt*	Output file *output.txt*
	8 4 5 1 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 3 2 0 0 1 1 2 0	4 2 0 1 5 4 2 0 1 5
	10 5 7 1 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 15 0 5 0 0 0 0 0 0 0 0 0 5 0 0 2 0 11 0 15 0 0 6 0 0 0 0 0 0 3 0 5 6 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0 10 0 12 11 7 1 5 2 3 2 12 24	5 1 3 8 9 4 0 2 6 No path exists.

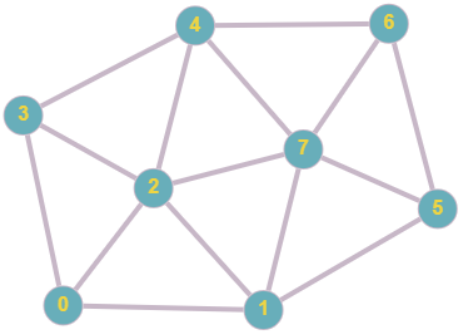
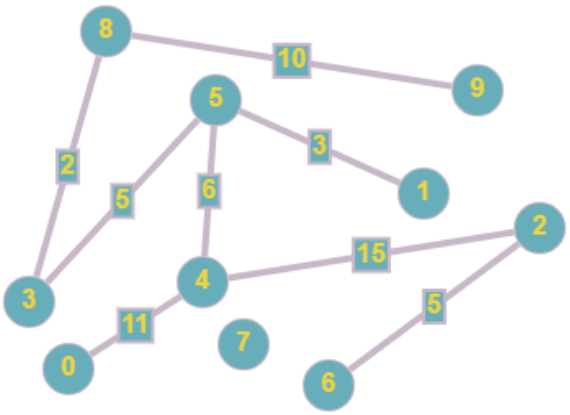
	14 13 12 1 0 0 0 0 11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 0 13 10 5 0 0 0 0 0 0 0 0 0 0 5 13 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 4 0 0 3 0 5 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 24 0 0 0 0 13 13 0 0 0 0 0 0 0 12 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 24 12 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 12 11 5 7 1 5 8 9 19 8 6 12 4 1	13 11 10 6 2 7 1 5 3 8 9 0 4 12 13 11 10 6 2 7 1 9 0 4 12

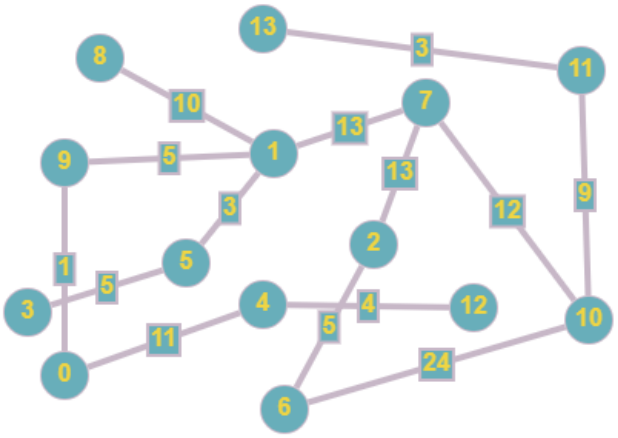
3. Uninform-cost search:

Graph	Input file *input.txt*	Output file *output.txt*
	8 4 5 2 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 3 2 0 0 1 1 2 0	4 2 3 6 7 0 1 5 4 6 5

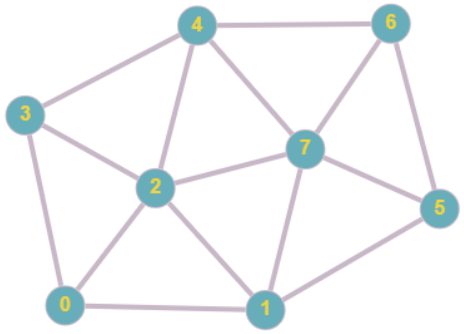
	10 5 7 2 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 15 0 5 0 0 0 0 0 0 0 0 5 0 0 2 0 11 0 15 0 0 6 0 0 0 0 0 3 0 5 6 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 10 0 0 0 0 0 0 0 0 10 0 12 11 7 1 5 2 3 2 12 24	5 1 3 4 8 9 0 2 6 No path exists.
	14 13 12 2 0 0 0 0 11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 3 0 13 10 5 0 0 0 0 0 0 0 0 0 0 5 13 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 4 0 0 3 0 5 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 24 0 0 0 0 13 13 0 0 0 0 0 0 0 12 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 24 12 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 12 11 5 7 1 5 8 9 19 8 6 12 4 1	13 11 10 7 6 1 2 5 9 0 3 8 4 12 13 11 10 7 1 9 0 4 12

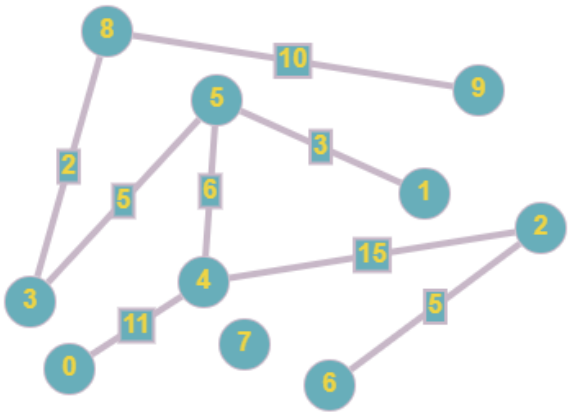
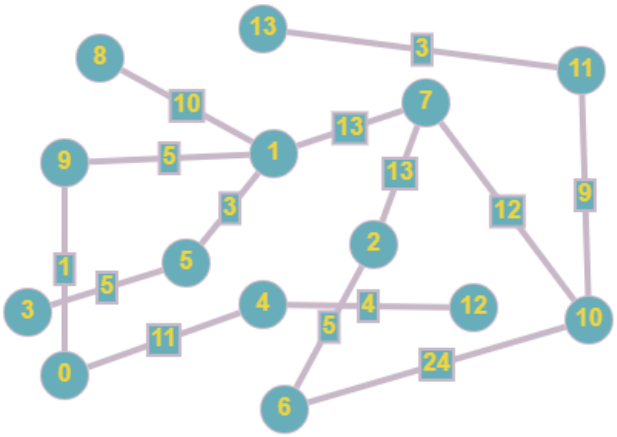
4. Iterative deepening search:

Graph	Input file *input.txt*	Output file *output.txt*
	8 4 5 3 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 3 2 0 0 1 1 2 0	4 4 2 3 6 7 4 2 0 1 3 7 3 0 2 6 5 4 6 5
	10 5 7 3 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 15 0 5 0 0 0 0 0 0 0 0 5 0 0 2 0 11 0 15 0 0 6 0 0 0 0 0 3 0 5 6 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 10 0 0 0 0 0 0 0 0 10 0 12 11 7 1 5 2 3 2 12 24	5 5 1 3 4 5 1 3 8 4 0 2 5 1 3 8 9 4 0 2 6 No path exists.

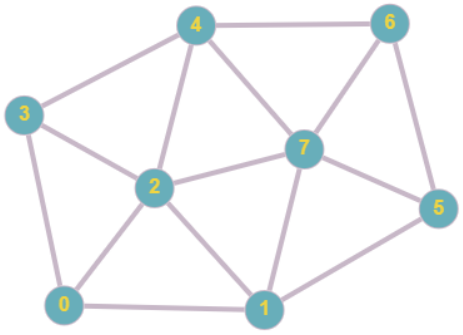
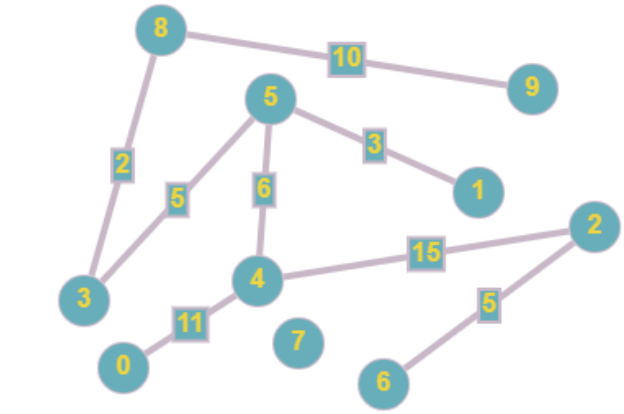
	14 13 12 3 0 0 0 0 11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 0 13 10 5 0 0 0 0 0 0 0 0 0 0 5 13 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 4 0 0 3 0 5 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 24 0 0 0 0 13 13 0 0 0 0 0 0 0 12 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 24 12 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 12 11 5 7 1 5 8 9 19 8 6 12 4 1	13 13 11 13 11 10 13 11 10 6 7 13 11 10 6 2 7 1 2 13 11 10 6 2 7 7 1 5 8 9 2 6 13 11 10 6 2 7 1 7 1 5 3 8 9 0 2 6 13 11 10 6 2 7 1 5 8 9 7 1 5 3 8 9 0 4 2 6 13 11 10 6 2 7 1 5 3 8 9 0 7 1 5 3 8 9 0 4 12 13 11 10 7 1 9 0 4 12

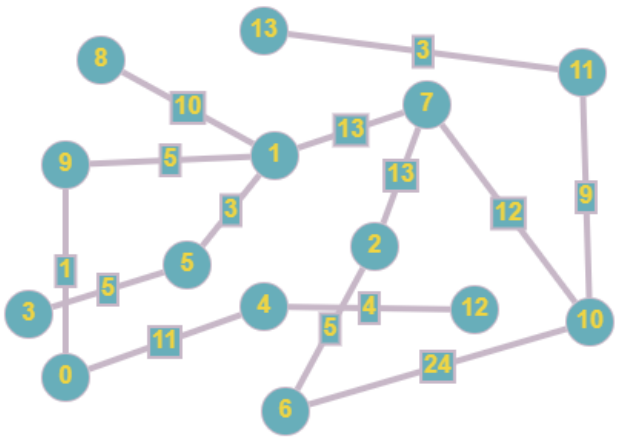
5. Greedy best first search:

Graph	Input file *input.txt*	Output file *output.txt*
	8 4 5 4 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 3 2 0 0 1 1 2 0	4 2 3 7 4 2 7 5

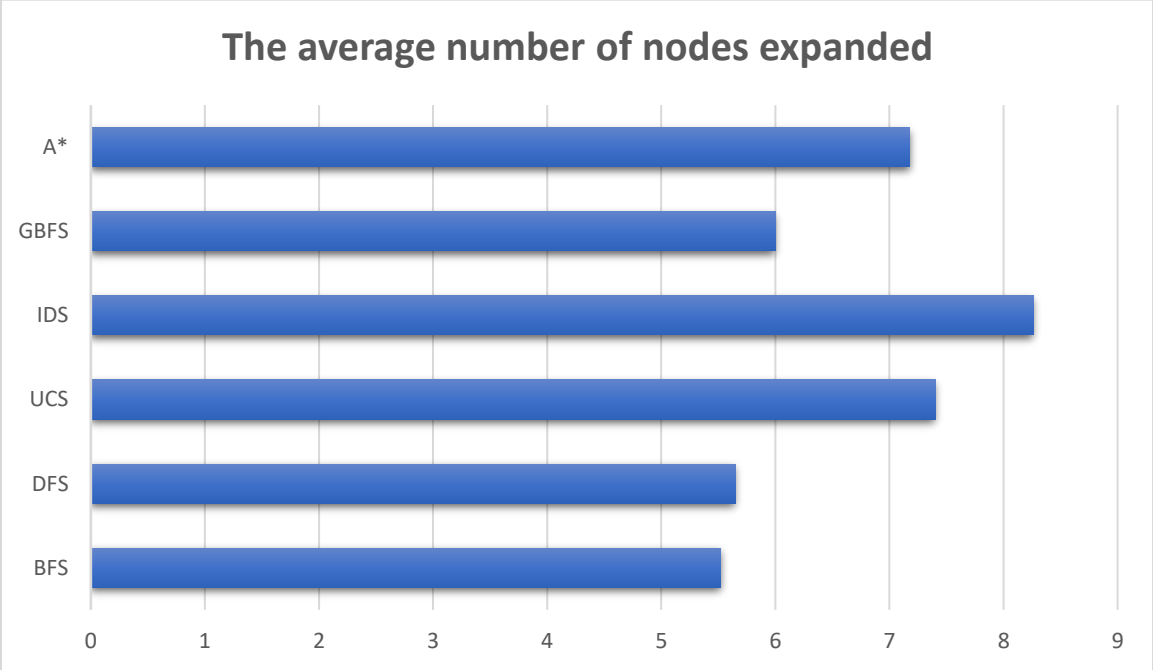
	10 5 7 4 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 15 0 5 0 0 0 0 0 0 0 0 5 0 0 2 0 11 0 15 0 0 6 0 0 0 0 0 3 0 5 6 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 10 0 0 0 0 0 0 0 0 10 0 12 11 7 1 5 2 3 2 12 24	5 3 4 2 6 1 8 0 9 No path exists.
	14 13 12 4 0 0 0 0 11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 0 13 10 5 0 0 0 0 0 0 0 0 0 0 5 13 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 4 0 0 3 0 5 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 24 0 0 0 0 13 13 0 0 0 0 0 0 0 12 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 24 12 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 12 11 5 7 1 5 8 9 19 8 6 12 4 1	13 11 10 6 2 7 1 5 3 9 0 4 13 11 10 6 2 7 1 9 0 4 12

6. Graph-search A*:

Graph	Input file *input.txt*	Output file *output.txt*
	8 4 5 5 0 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 3 2 0 0 1 1 2 0	4 2 3 7 5 4 7 5
	10 5 7 5 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 15 0 5 0 0 0 0 0 0 0 0 5 0 0 2 0 11 0 15 0 0 6 0 0 0 0 0 3 0 5 6 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 10 0 0 0 0 0 0 0 0 10 0 12 11 7 1 5 2 3 2 12 24	5 3 4 1 8 2 0 6 9 No path exists.

	14 13 12 5 0 0 0 0 11 0 0 0 0 1 0 0 0 0 0 0 0 0 0 3 0 13 10 5 0 0 0 0 0 0 0 0 0 0 5 13 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 4 0 0 3 0 5 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 24 0 0 0 0 13 13 0 0 0 0 0 0 0 12 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 24 12 0 0 0 9 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 12 11 5 7 1 5 8 9 19 8 6 12 4 1	13 11 10 7 2 6 1 5 9 3 0 4 12 13 11 10 7 1 9 0 4 12

III. Summary:



- Most algorithms reached more than half of its nodes.
- IDS algorithm performs huge expanded nodes in my 3rd test case. It is dominance on other algorithms to receives the worst in case of finding a pair of nodes which so far away bacause of the requirement of traversing by depth.

- A* is the most suitable and performance algorithm. *(At least in my pack of test cases)*. Because of heuristic values.
- BFS and DFS are similar in the average number of nodes expanded. But BFS is more suitable when applied to search the destination closer to the source, but DFS is more suitable in the other hand.

IV. References:

Online graph creator.
Kartiikthakur Artificial Intelligence searches algorithms.
Dr. Nguyễn Ngọc Thảo Shared Google Drive.