Artificial Intelligence

Search Strategies

Task Report 2

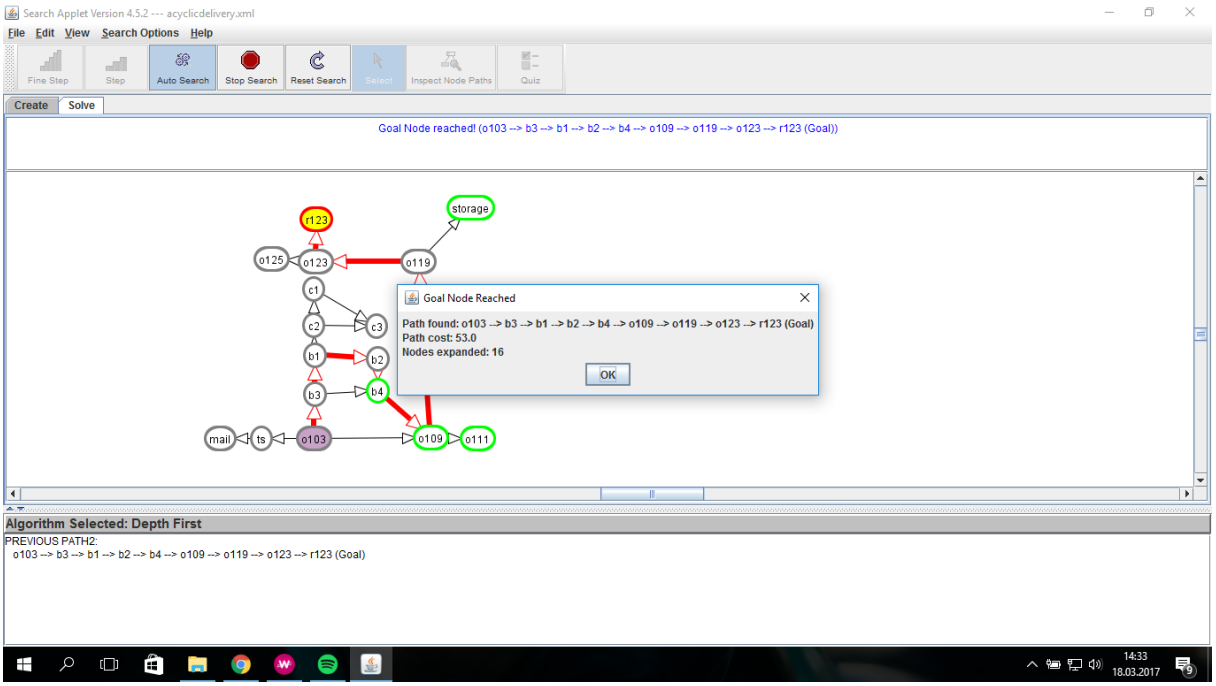
Harun GÜLEÇ

Bialystok,2017

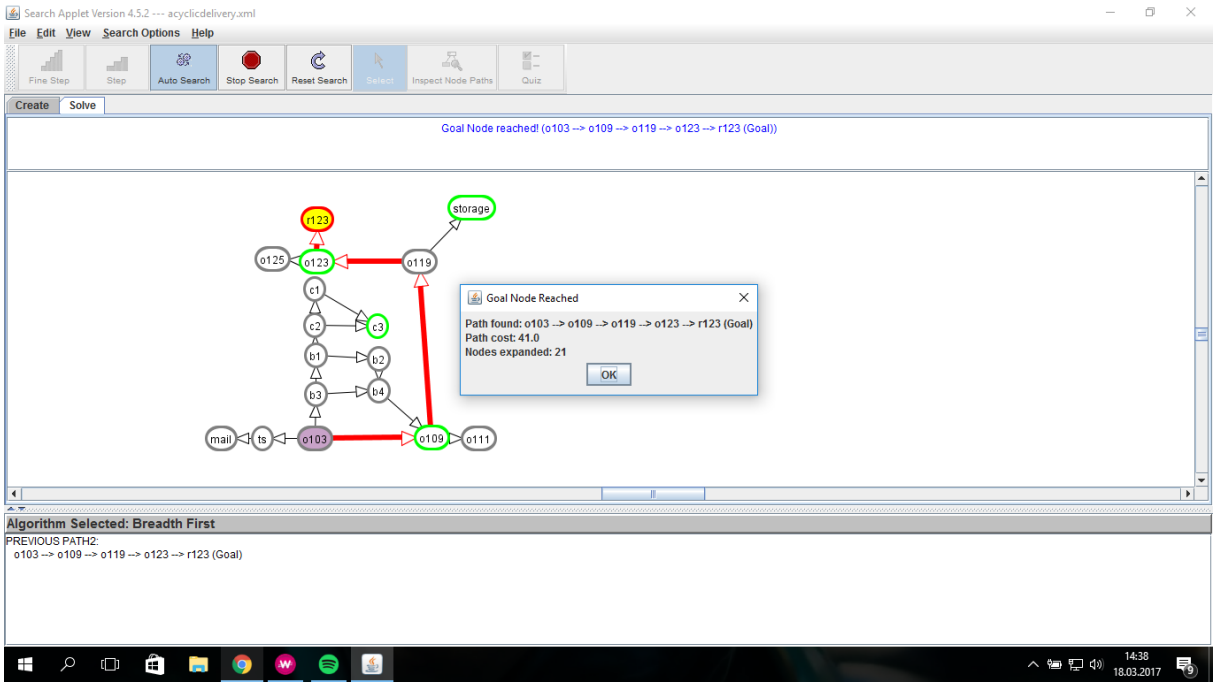
Task 1

At this task our main object is comparing 6 different search algorithms. To do that we used *Delivery robot (acyclic)* sample data that presented in AIspace software and the data is about problem of the transport of elements.

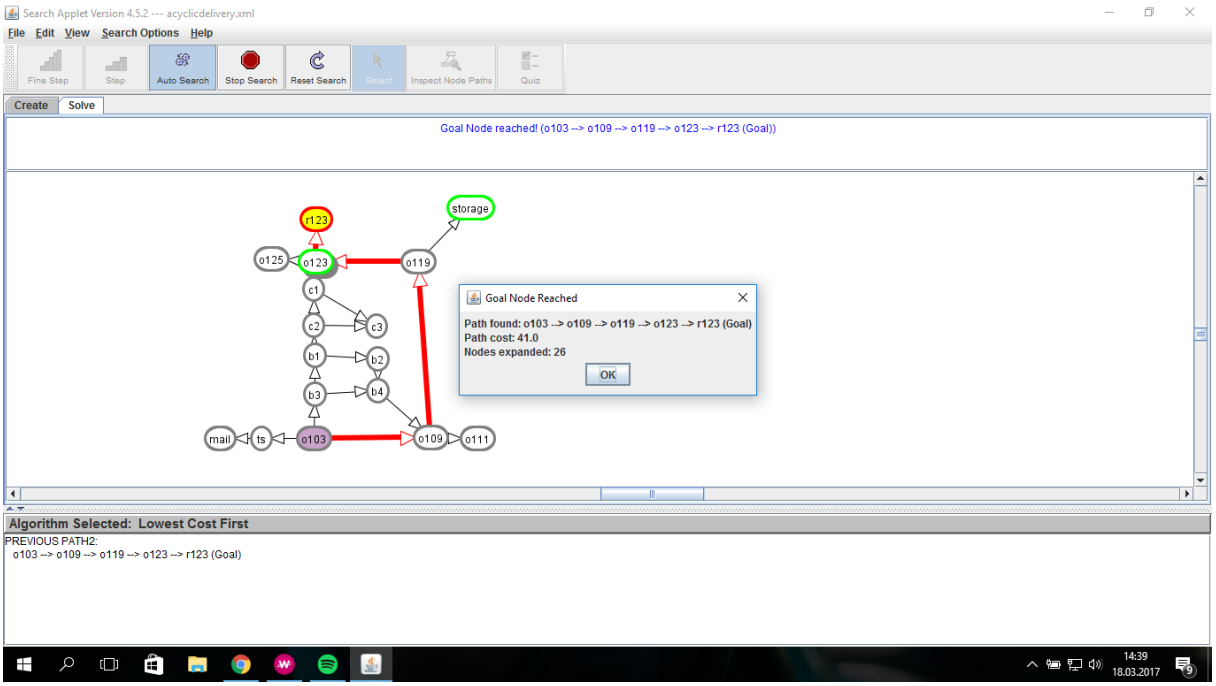
First of all, we use depth first search algorithm which searching as far as possible along each branch before backtracking. We find the goal node very quickly. Cost is 53.0 and expanded nodes 16.



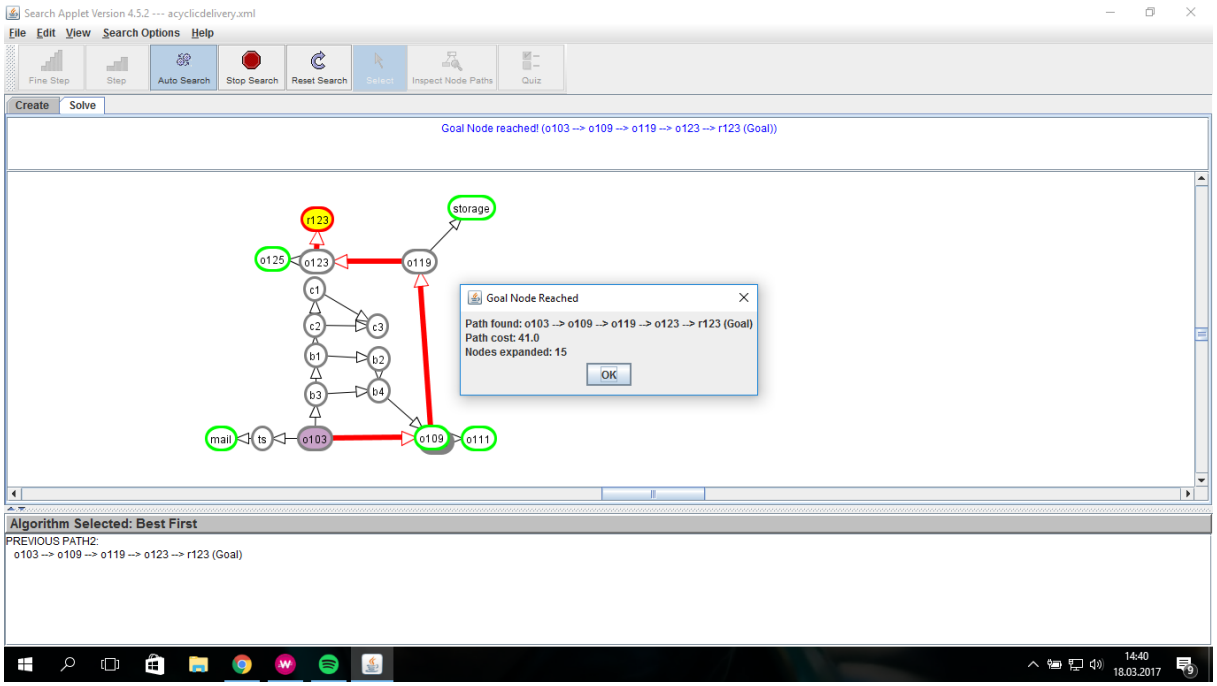
Secondly, we tried breadth first search algorithm which searching the neighbor nodes first, before moving to the next level neighbors. Search time is a little bit longer than DFS. But cost is better. Cost is 41.0 and expanded nodes 21.



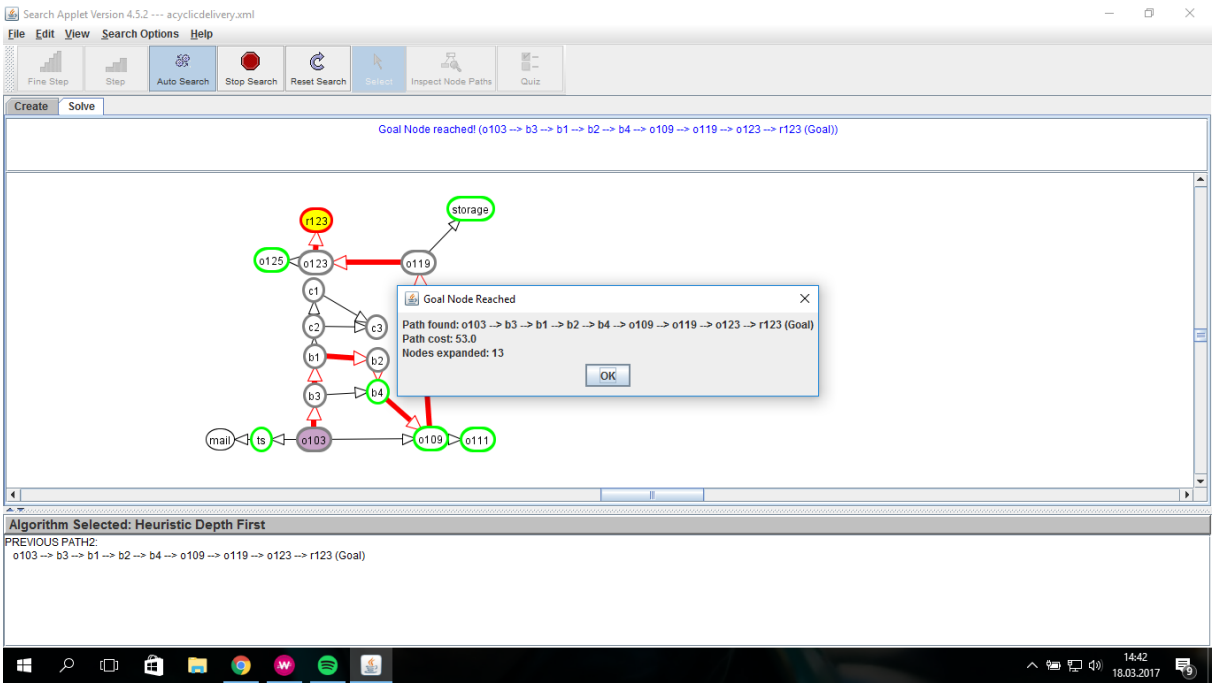
After, we use lowest cost first algorithm that searching lowest cost nodes first among the expanded nodes. Search time is long but it can find lowest cost path. Cost is 41.0 and expanded nodes 26.



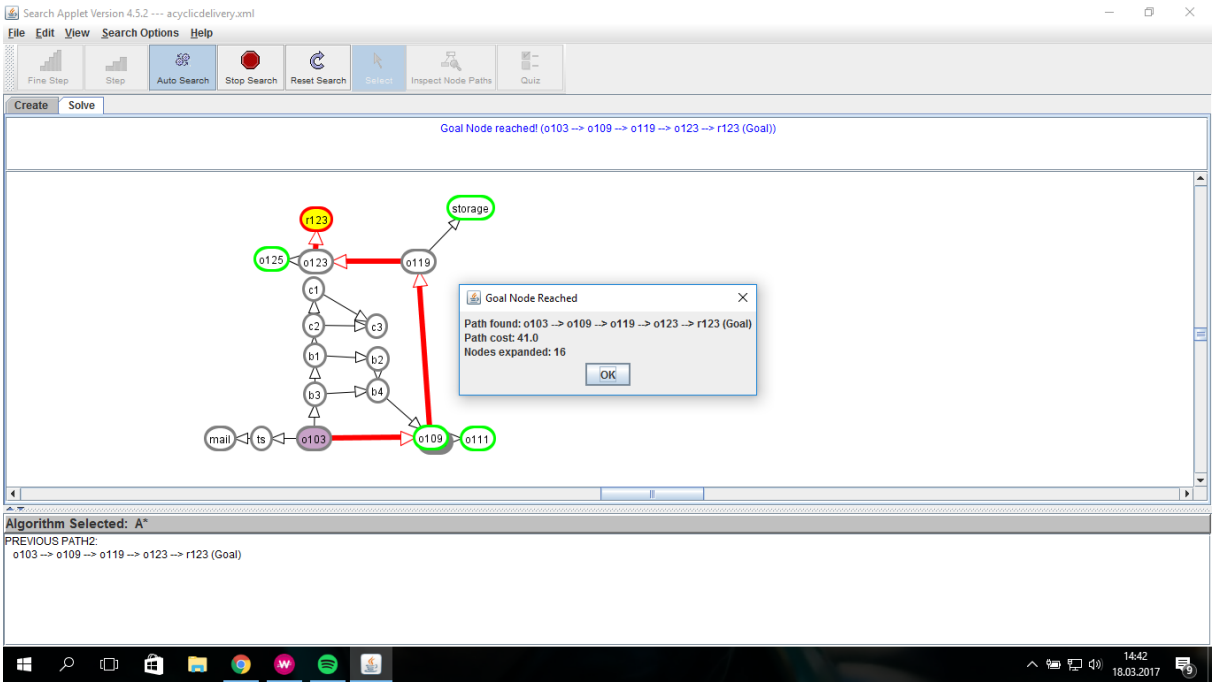
After that, we check the result of best first algorithm which a graph by expanding the most promising node chosen according to a specified rule. In our example nodes have heuristic values according to proximity. We can find low cost path quickly. Cost is 41 and expanded nodes 15. It means we find lowest cost with minimal searching time.



After, we used heuristic depth first search algorithm which searching like depth first but select low heuristic branch firstly. This time we have shortest searching time but cost is not minimal. The cost is 53.0 and expanded nodes 13.



Finally, we used A* algorithm which searching among all possible paths to the solution (goal) for the one that incurs the smallest cost. Searching time is a little bit long but not so much because it searches possible paths using heuristic values. It can find lowest cost path in similar time with best first search.



Task 2

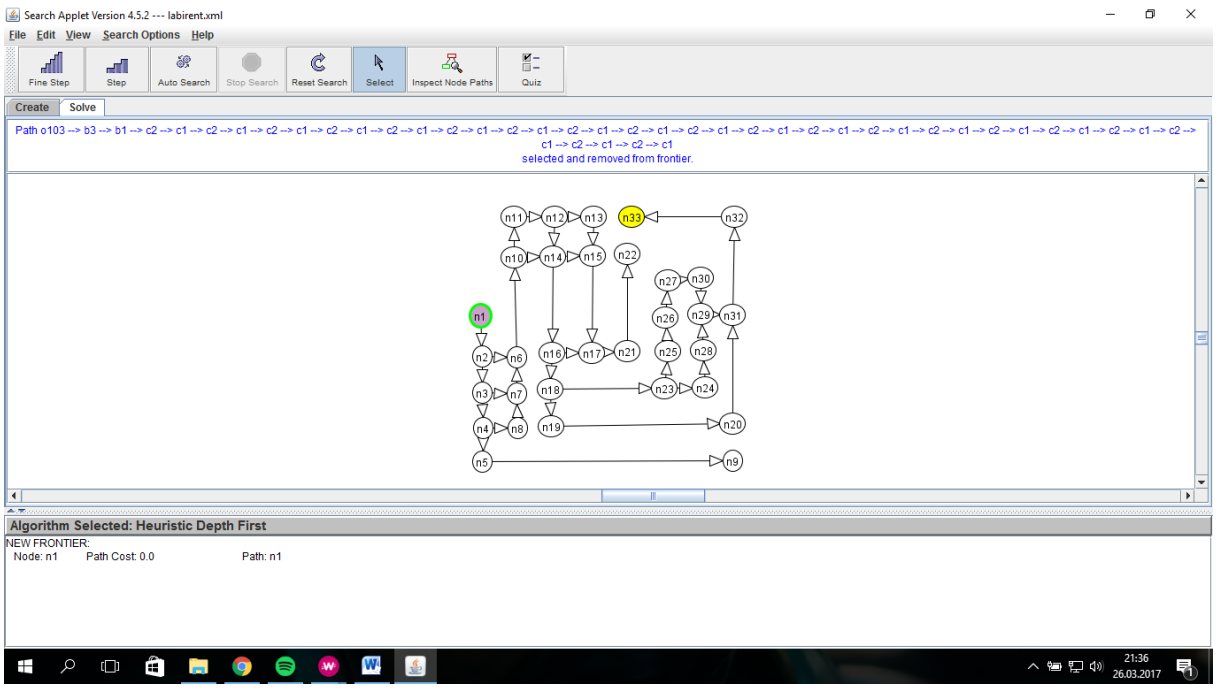
At this task our main object is comparing 6 different search algorithms on a cyclic path. To do that we used *Delivery robot (cyclic)* sample data that presented in AIspace software and the data is about problem of the transport of elements.

We check all algorithms' results and we can say clearly without loop detection only Breadth first algorithm works very well. A* algorithm and lowest cost first algorithm work very slowly.

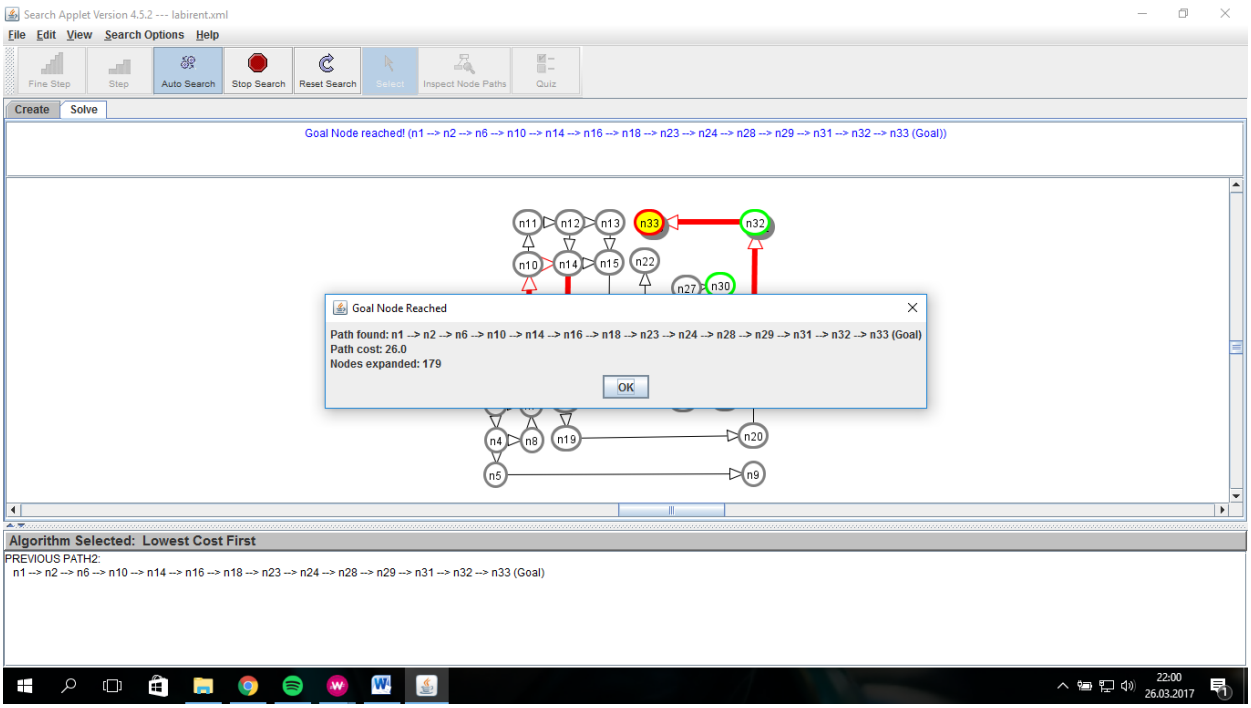
After that we enabled loop detection option, we got results almost the same as previous results.

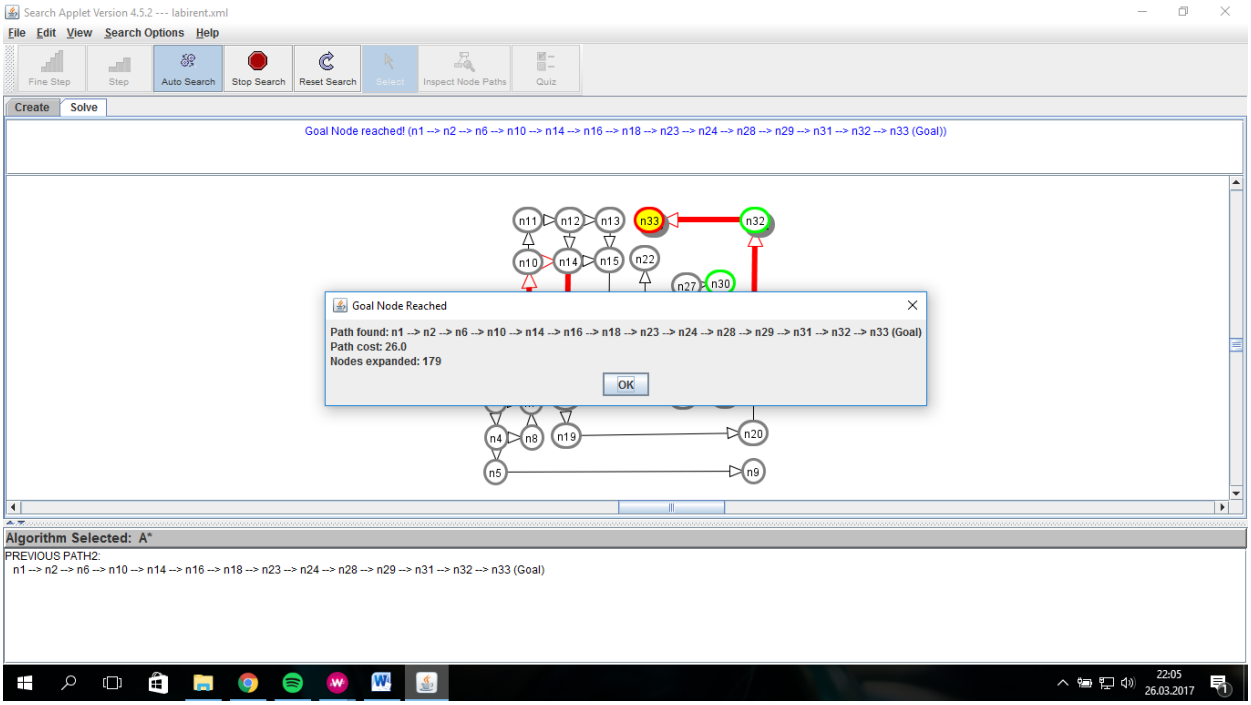
Task 3

At this task our main object is construct a graph representing the maze and check which algorithm can find shortest path. To do this, we created a graph represent the maze as shown below.



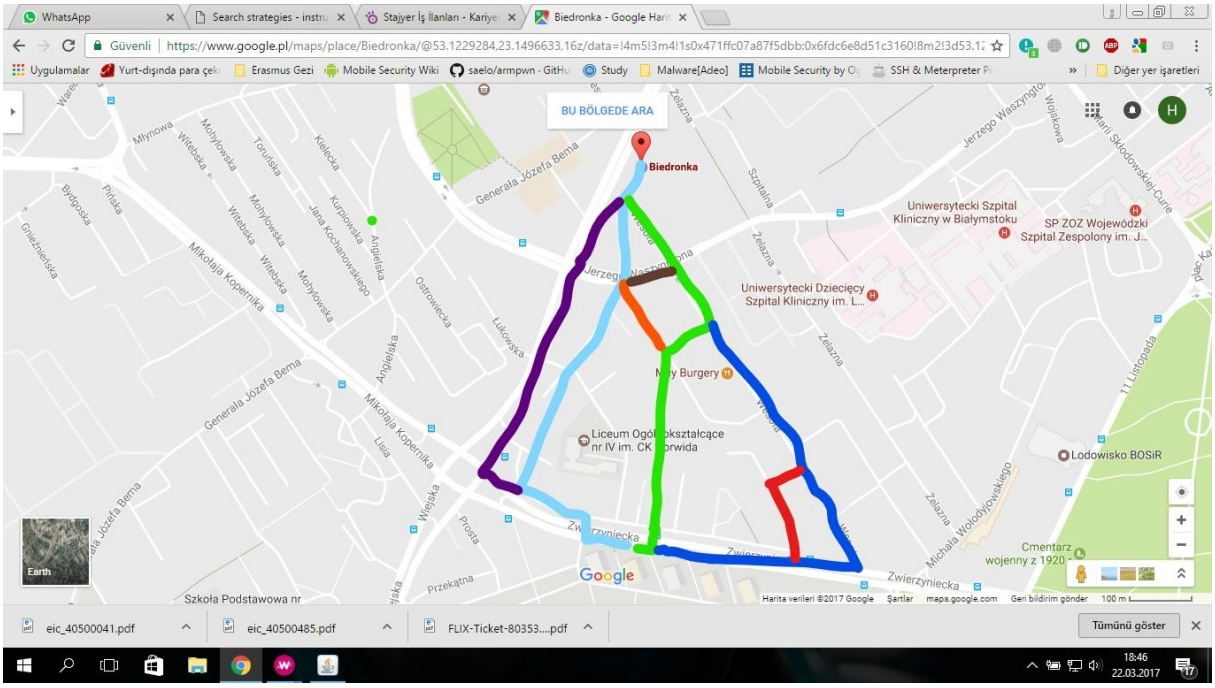
Lowest cost first algorithm and A* algorithm found the shortest path as 26.0 path length.



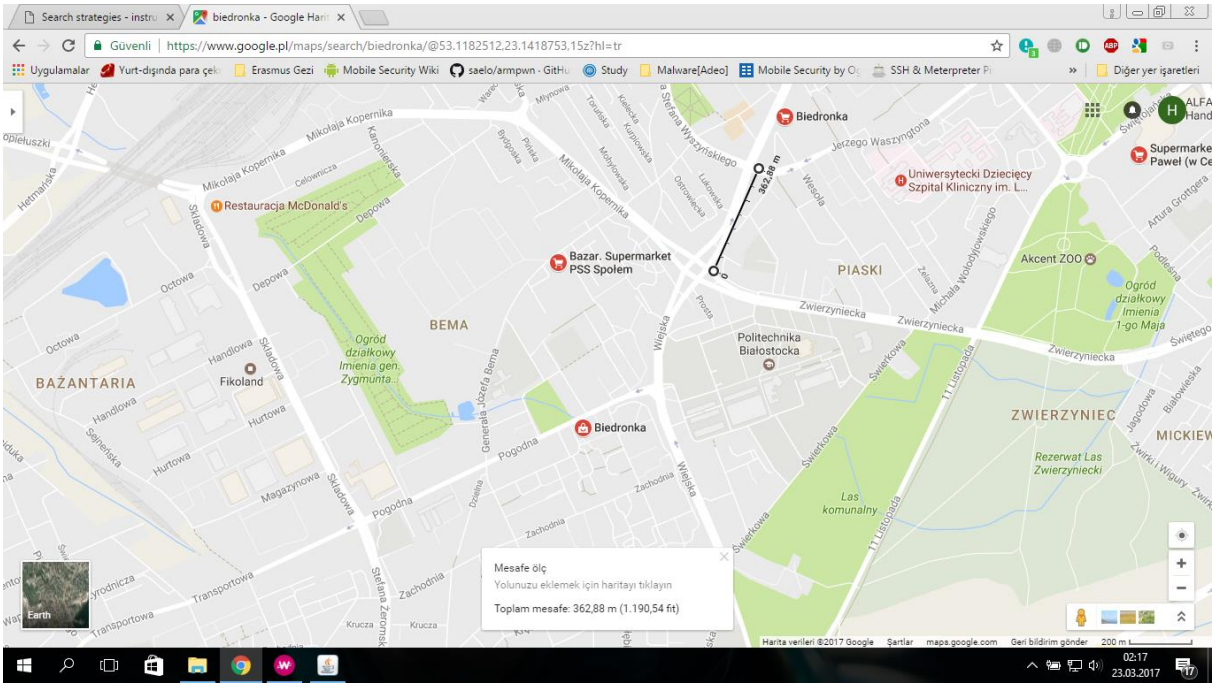


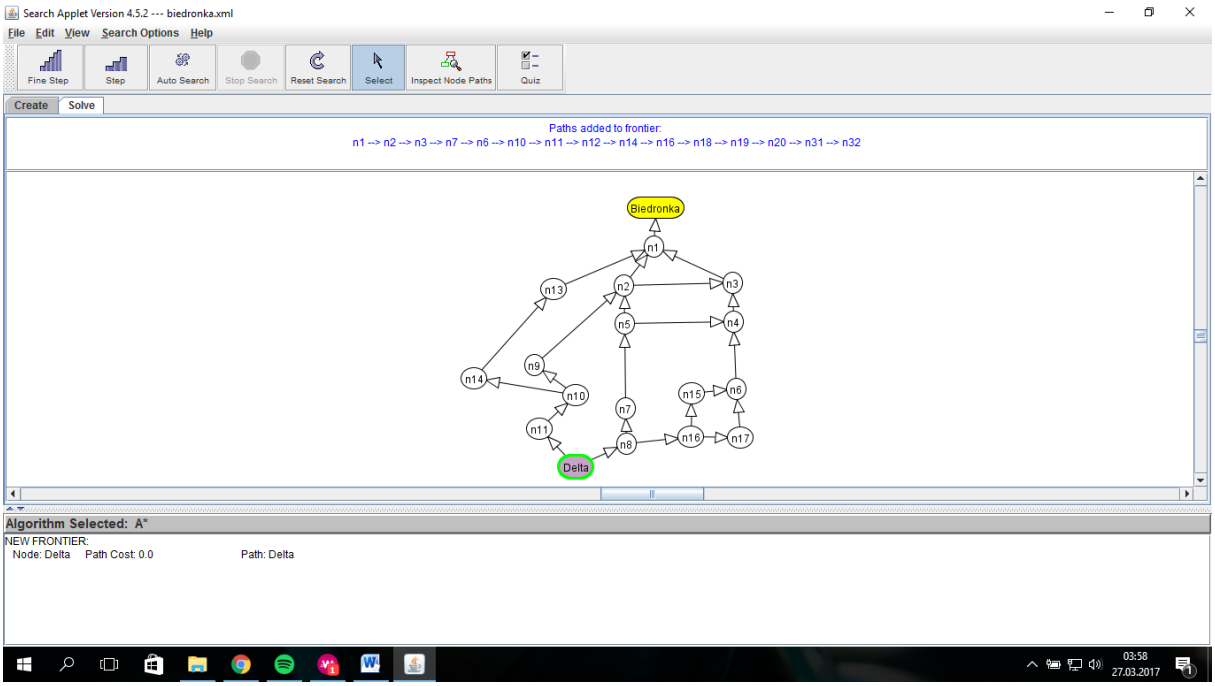
Task 4

At this task, we have to propose a solution to problem using search algorithms. I selected *Shortest Path Akademik Delta to Biedronka* as problem. I used Google Maps to determine nodes. I draw all paths that we use going to Biedronka as shown below.

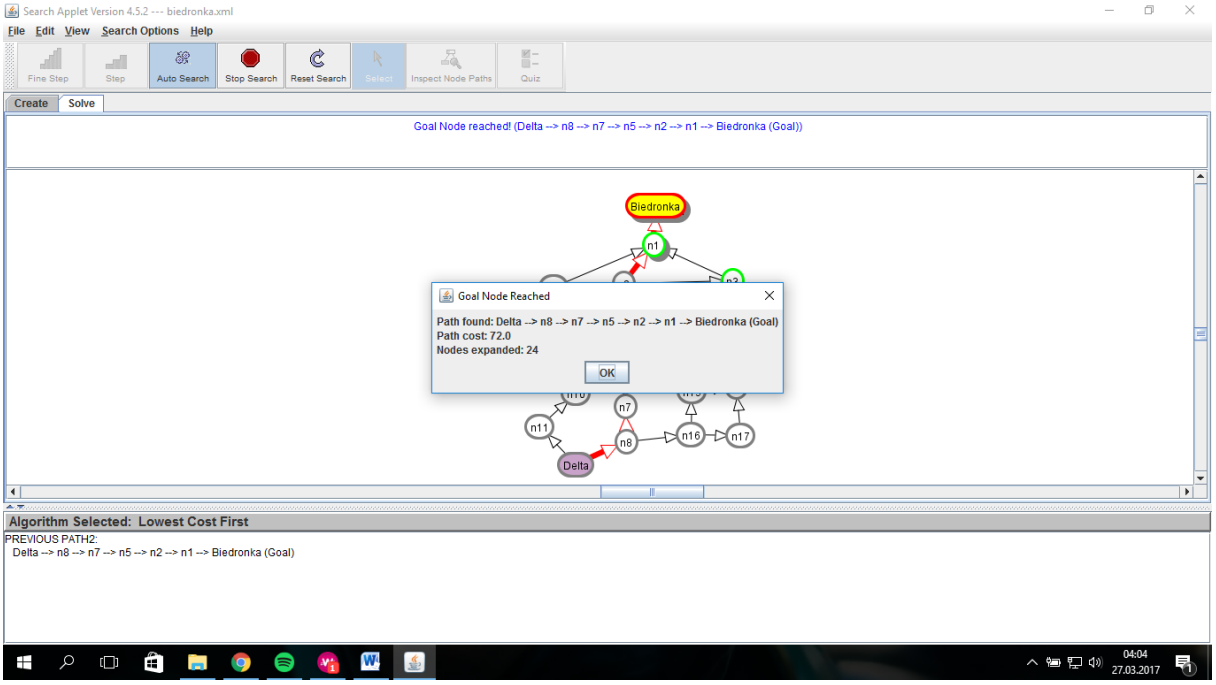


After that, I created graph diagram according to these paths. To calculate edges' weight I used Google Maps again as shown below. Distance between two nodes divided by 10 is weight of edges. Also, traffic lights' cost is +5 point.





When we use lowest cost search algorithm to find shortest path to biedronka, cost is 72.0 as shown below.



In conclusion, we can use depth first search algorithm when we have to find an object quickly. If we have a rule to choose next node, we can use best first or heuristic depth first search algorithm. When we need to find shortest path, cheaper transportation etc. we can use lowest cost first algorithm. In such a complex situation as library searching, we can use A* algorithm to find an object easily and quickly.