* **Intelligent robot**
* We can see the maze as a tree. In each step we can see the junction (even if there is only way to go straight) as the node in the tree. So we can use the tree search method. Firstly we use blinding search. There are also some sub kinds in blinding search: branch first search and depth first search. While in our situation, the size of the maze is fixed (or said it is small compare to infinite), and we cannot jump from one node to another reached node in one step, BFS is not suitable. Then we choose depth first search. The left hand robot and right hand robot are both depth first search and representative are preorder and postorder. Actually in this situation (the origin is the left top point and the destination is right bottom) they work well. While when we change the origin or the destination, they will work badly because of the depth first search cannot find the optimal result. Then I though to improve the method use the branch of depth search method called Iterative Deepening Search. However as I said before we cannot jump from one node to another reached node in one step. So this method is not suitable either. Then is use random run but record the walk path to improve the intelligent. This method is also the depth first search but no fix order. The advantage is in each junction that contain the correct the way, this method is possible to go into the right way, while either one of left hand robot or right hand robot will definitely go into the wrong way. So the float of this method is huge, it may find the right way in very short time, or… Then I thought that I couldn’t improve it in blinding, so I use heuristic search. The information I get is the destination of the maze. For each time the robot run, it will try to find the way that will make itself closer to the destination. The greedy method let the robot seems likely to go on the right way. Although some times the path in maze is at opposite poles. In most time it will find the correct direction. In those file that you have provided, the method preformed good, which at most time is quicker than any other method. And after I changing the destination into the center of the maze, it performs better.

|  |  |  |  |
| --- | --- | --- | --- |
| * Maze | * Left hand | * Right hand | * Greedy |
| * Hak-20x20 | * 116 | * 626 | * 196 |
| * Prim-15x15 | * 210 | * 238 | * 56 |
| * Prim-20x30 | * 558 | * 640 | * 240 |
| * Rec-15x10 | * 65 | * 231 | * 167 |

* As you can see that the greedy perform will in the most time