

In this homework, I compared four search algorithms: Dijkstra's, A\*, ARA and ANA.  
Run with: `python main.py (choose one from: d, a, ara, ana) (choose one from: the difficulties given, test1, test2)`

Comparison:

All of the algorithms can give an optimal solution eventually. ARA and ANA would give an suboptimal solution first, then improve on that.

1. Dijkstra's algorithm: very conservative on which next step to take, always explores all the possible path before settle on one choice.
2. A\* algorithm: performs well in this 2d case. This is because the  $h(s)$  given (the steps counts to the goal) is a very accurate estimate of the cost to the goal.
3. ARA: the algorithm performance depends heavily on the  $\epsilon$  and  $\delta_\epsilon$  of our choice.
4. ANA: can converge fast enough, without a given  $\delta_\epsilon$ .

Run time table (unit: s):

	trivial.gif	medium.gif	hard.gif	very_hard.gif	test1.png	test2.png
Dijkstra's	0.040367	0.56063	2.41201	19.72411	0.13627	0.13097
A*	0.040688	0.65363	2.20196	19.36601	0.51285	0.21613
ARA	0.04208	0.48801	1.61464	13.95291	0.03553	0.18065
ANA	0.03811	0.45619	1.78904	13.63570	0.07356	0.16949

Overall, ARA and ANA are faster than Dijkstra and A\*, especially in larger maps.