Analysis and Comparison

- Find a map size (dim) that is large enough to produce maps that require some work to solve, but small enough that you can run each algorithm multiple times for a range of possible p values. *How did you pick a* **dim***?"*

- For p 0.2 generate a solvable map, and show the paths returned for each algorithm. *Do the results make sense? ASCII printouts are fine, gut good visualizations are a bonus*

- Given **dim** how does maze-solvability depend on p? For a range of p values, estimate the probability that a maze will be solvable by generating multiple mazes and checking them for solvability. What is the best algorithm to use here? Plot *density bs solvability,* and try to identify as accurately as you can the threshold p_0 where for $p < p_0$, most mazes are solvable , but $p > p_0$, most mazes are not solvable

- For p in $[\,0, p_0]$ as above, estimate the average or expected length of the shortest path from start to goal. You may discard unsolvable maps. Plot *density vs expected shortest path length* what algorithm is most useful here?

- Is one heuristic uniformly better than the other for running A*? How can they be compared? Plot the relevant data and justify your conclusions

- Do these algorithms behave as they should?

- For DFS, can you improve the performance of the algorithm by choosing what order to load the neighboring rooms into the fringe? What neighbors are 'worth' looking at before others? Be thorough and justify yourself.

- On the same map, are there ever nodes that BD-BFS expands that A* doesn't? Why or why not? Give an example, and justify