A\*

* History
  + Optimized version of Dijkstra
  + Dijkstra created in 1956 (published in 1959), A\* in 1968
  + Created to help a robot pathfind
    - Robot was using Graph Traverser, which ignores g(n) in Dijkstra, and only focuses on the heuristic h(n)
    - A\* put the g(n) and the h(n) together
  + Even though it is the successor, a lot of places refer to Dijkstra as, “A special case of A\* where h(n) is always zero”
* Examples
  + Dijkstra, only when you add the values to the priority queue, it isn’t just the value up to that point, but also the heuristic, which approximates how much further you have to go. Note: the heuristic should always be an underprediction.
  + Pull some gifs and others from websites
* Walkthrough
  + Pull geeks for geeks A\* code, then code the example maps myself, along with displaying the output (assuming they don’t display output)
  + Further Reading:
    - <https://briangrinstead.com/blog/astar-search-algorithm-in-javascript/>
      * Contains an interactive web feature to show A\* getting from point A to point B
    - <https://www.simplifiedpython.net/a-star-algorithm-python-tutorial/>
  + References:
    - <https://www.geeksforgeeks.org/a-search-algorithm/>
      * For Code, info, plus some images
    - [https://en.wikipedia.org/wiki/A\*\_search\_algorithm#cite\_note-5](https://en.wikipedia.org/wiki/A*_search_algorithm#cite_note-5)
      * For Gifs
    - <https://gist.github.com/OmarElGabry/d2670245b167d874eb2846913901066a>
      * For Code, complete A\* implementation
    - <https://www.youtube.com/watch?v=g024lzsknDo&ab_channel=KevinWang>
      * For A\* versus Dijkstra comparison gif
    - <https://medium.com/swlh/dijkstras-algorithm-in-an-undirected-graph-c0c086d77145>
      * For example image
  + Github Repo Link