**1/17/18**

03-Basic-Search-Algorithms

**Heuristics**

* A **heuristic evaluation function**, in state-space search, is a function h: Σ →ℜ+ that can be used as an estimate of how close a state is to a goal or simply to prioritize states for attention
  + ex.: hot and cold, euclidian distance between city and goal in routing problem.

**Best-First Search**

* Provided we have a heuristic evaluation function, we can prioritize states for expansion using the function
  + use priority queue to implement OPEN list using heuristic value
* greedy algorithm

**Ideal Distances in A\* Search**

* also a BFS, but also uses a g value (uniform cost)
  + g(s) = value to get from start to s node
  + h(s) = cost to get from s to goal node
* f(s) = cost/distance of shortest path from start node through s node and end at goal node
  + f(s) = g(s) + h(s)

**Estimated Distances in A\* Search**

* Let g’(s) be an estimate of g(s) based on the **currently** known shortest distance from the start node to s.
* Let the h’(s) be a heuristic evaluation function that estimates the distance (path length) from s to the nearest goal node.
* f’(s) = g’(s) + h’(s)
* Best-first search using f’(s) as the evaluation function is called **A\* search.**

**Admissibility of A\* Search**

* Under certain conditions, A\* search will always reach a goal node and be able to identify a shortest path to that node as soon as it arrives there.
* Conditions
  + hʹ(s) must not exceed h(s) for any s.
  + w(si , sj ) > 0 for all si and sj .