BFS:

* Good news:
  + Complete
  + Guaranteed to ﬁnd the shallowest path to the goal

This is not necessarily the best path! But we can use Priority queue instead of queue

* + Diﬀerent start-goal combinations can be explored at the same time
* Bad news:
  + Exponential time complexity:
  + This is the same for all uninformed search methods
  + Exponential memory requirements!

DFS:

* Good news:
  + Space complexity O(bd) (why?)
  + It is easy to implement recursively (do not even need a queue data structure)
  + More eﬃcient than BFS if there are many paths leading to a solution.
* Bad news:
  + Exponential time complexity:

This is the same for all uninformed search methods

* + Not optimal
  + DFS may not complete! (Due to the max search depth limit)
  + NEVER use DFS if you suspect a big tree depth

Admissible heuristics → Consistent if

Properties of A∗

* Complete!
* Optimal!
* Exponential worst-case time and space complexity (why?)
  + But with a perfect heuristic, the complexity is O(bd), because we would only expand the nodes along the optimal path
  + With a good heuristic complexity is often sub-exponential
* Optimally eﬃcient: with a given h, no other search algorithm will be able to expand fewer nodes