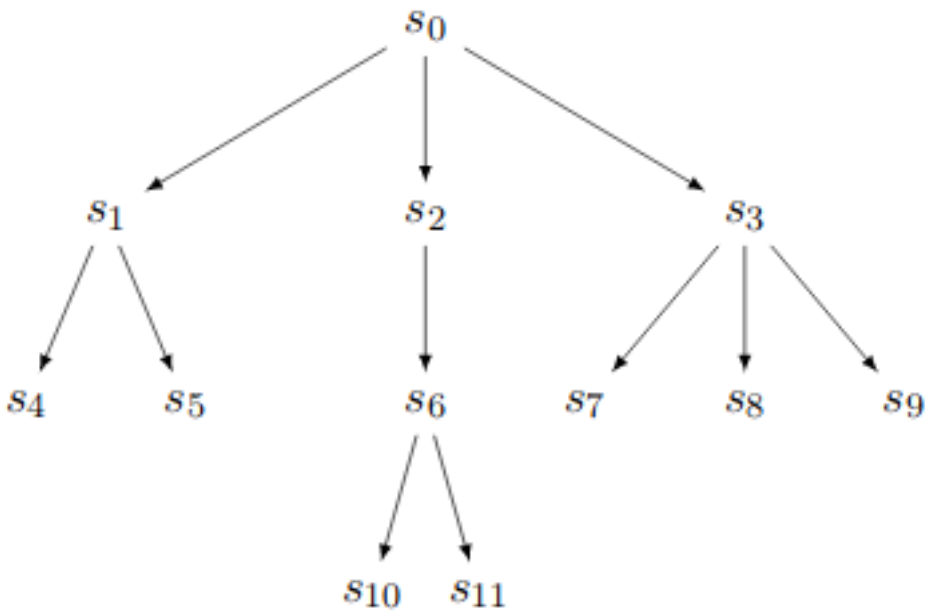
# AI Tutorial 3

Brandon Skerritt

December 31, 2017




Questions 1 & 2 refer to the above image

## 1 Question 1

Suppose A is a search algorithm. Assume A performs a depth 2 limited search strategy. Give a (possible) sequence in which the states are selected by A.

### 1.1 Answer

s0s1s4s5s2s6s3s7s8s9

## 2 Question 2

Assume A performs an iterative deepening strategy. Give a (possible) sequence in which the states are selected by A.

### 2.1 Answer

s0s1s2s3s4s5s6s7s8s9s10s11

## 3 Question 3

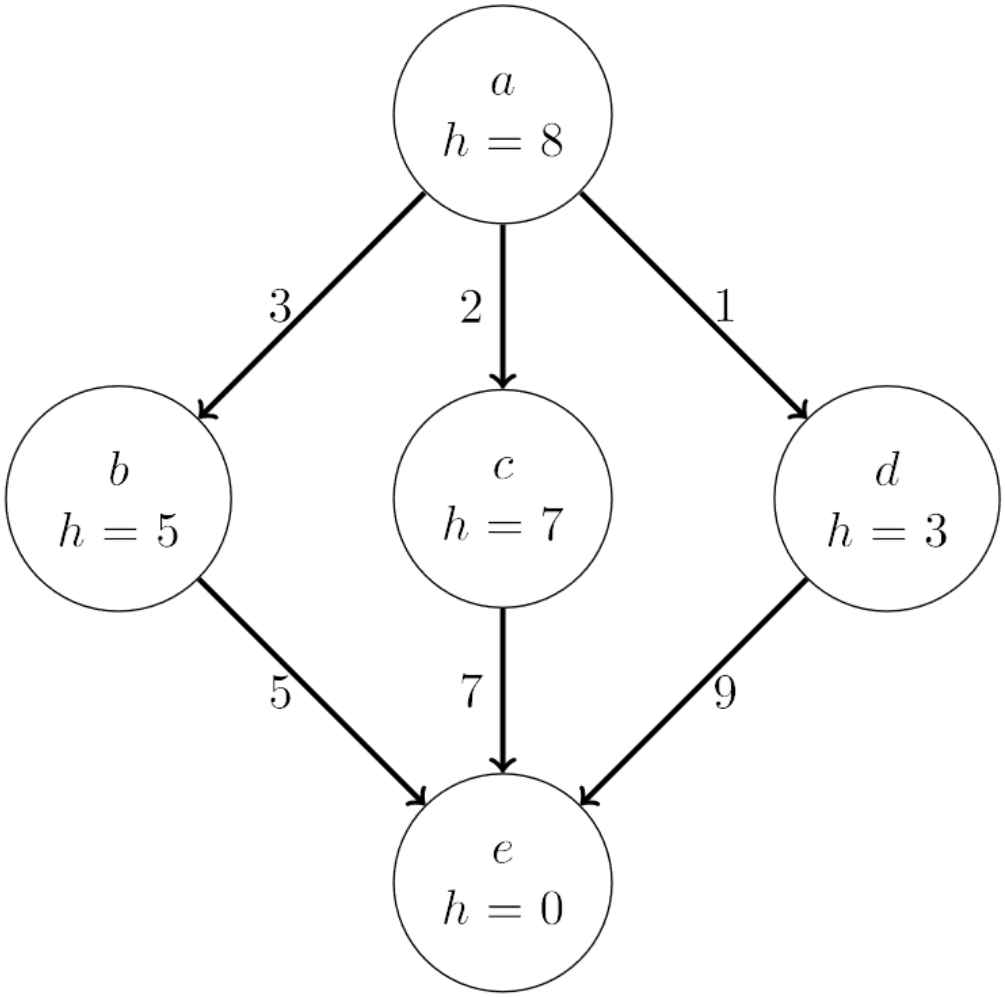
Determine the space complexity of iterative deepening: given the branching factor b of the search tree and the depth d of the short-est path to a goal state, what is the maximal number of paths in the frontier?

### 3.1 Answer

A depth $c$ limited search is performed for every c <= d (for every level below or equal to d) thus the frontier size is bc, thus it is bd at the worst-possible case.

## 4 Question 4

Consider the following search graph with states {a, b, c, d, e}, start states start=a, and a single goal state e. The heuristic value for each state is expressed within the state underneath the state name, and the cost of each action is shown next to the arrow representing the action.




## 5 Question 5

Assume a uniform cost algorithm is applied to the search graph. Give a sequence of frontiers that could be computed by the algorithm, there turned path and its cost.

### 5.1 Answer

A uniform cost search is one that uses path costs. So the algorithm will look like
start at a
a not goal, frontier = {b:3, d:1, c:2}
d not goal, frontier = {b:3, de:10, c:2}
c not goal, frontier = {b:3, de:10, ce: 9}
b not goal, frontier = {de:10, ce:9, be:8}

be goal, frontier = {de:10, ce:9} Always make full path names, always include the frontier once goal is found.

# 6 Question 6

Assume a greedy algorithm is applied to the search graph. Give a sequence of frontiers that could be computed by the algorithm, there turned path and its cost

## 6.1 Answers

A greedy search uses heuristics so
a:8
a not goal, frontier = {b:5, c:7, d:3}
d not goal, frontier = {b:5, c:7, de:0}
de goal, frontier = {b:5, c:7} Note: With heuristic, you only use the heuristic in the node. You don't add them together or anything.

# 7 Question 7

Assume an A* algorithm is applied to the search graph. Give a sequence of frontiers that could be computed by the algorithm, the returned path and its cost.

## 7.1 Answer

A* search uses heutistic + cost search
a
a not goal, frontier = {b:8, c:9, d:4}
d not goal, frontier = {b:8, c:9, de:10}
b not goal, frontier = {c:9, de:10, be:8}
be goal Important to note that cost value of each line is added up to the end by heuristic is only added of node it is traveling to