Search algorithms

- Initial state
- Goal state
- $b$: branching factor

- $m$: max depth

Features of search algorithms

- Time complexity:
  - How long does it take the algorithm to find a goal that is at the frst right at level $d$?

- Optimality:
  - Is the algorithm guaranteed to find the optimal (lowest cost) goal?

- Space complexity:
  - Size of the frontier (or main storage)

DFS:
- $O(b^d)$

BFS:
- $O(b^d)$

DFS vs. BFS:
- DFS $X$
- BFS $V$

DFS = $b \times m$
BFS = $b^d$
Completeness: if the goal node is reachable from the start node, is the algorithm guaranteed to find the goal?

- BFS
  - yes
- DFS
  - loops \( \Rightarrow \) infinite search
  - \( \Rightarrow \) no

yes only if \( m \) is finite

DFS  BFS  2
a)  1  1
b)  1  0
  4
c)  0  0
  2
d)  0

e) idle
f) I do care but not right now
  
g) still thinking

\[ \text{an} \]
depth limited search

DFS with depth limit = $l$

a node cannot generate children if it is at level $l$

pick up where you left off

iterative deepening search (IDS)

for $l = 1$ to $\infty$

do DFS with depth limit $l$

time complexity

space complexity $b \times l$ ✓

optimality yes ✓

completeness yes ✓
\( \frac{1}{5} \)
\( \frac{1}{4} \)
\( \frac{1}{2} \)
\( \frac{1}{3} \)
\( \frac{1}{5} \)
\( \frac{1}{6} \)
\( \frac{1}{6} \)
\( \frac{1}{6} \)
\( \frac{1}{6} \)
\( \frac{1}{6} \)

\( (b^d)! \)

\[ b^1 + b^2 + b^3 + b^4 + b^5 \]

\[ 56^1 + 4b^2 + 3b^3 + 2b^4 + b^5 \]

run 5 times.

\[ d(b) + (d-1)b^2 + \ldots + 1(b^d) \]

\[ O(b^d) \sim O(b^d) \]

BFS

d = 5
b = 10

5 0 + 4 0 0 + 3 000 + 20,000
+ 100,000

= 123,450

10 + 100 + 1,000 + 10,000
+ 100,000

= 111,110
Read the solution up to parent?

store
- actions taken
- path cost

world only work for DFS