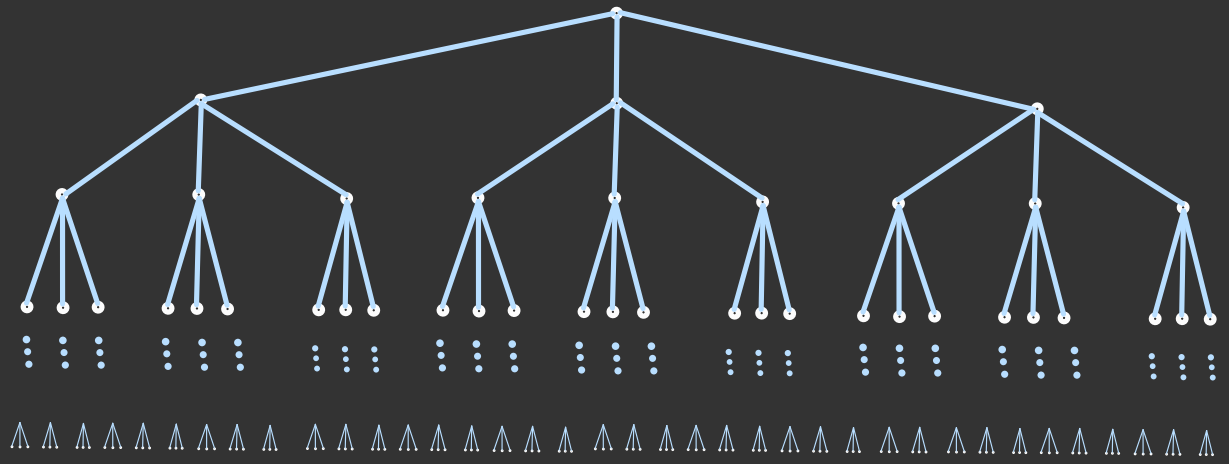
runtime of bfs and dfs

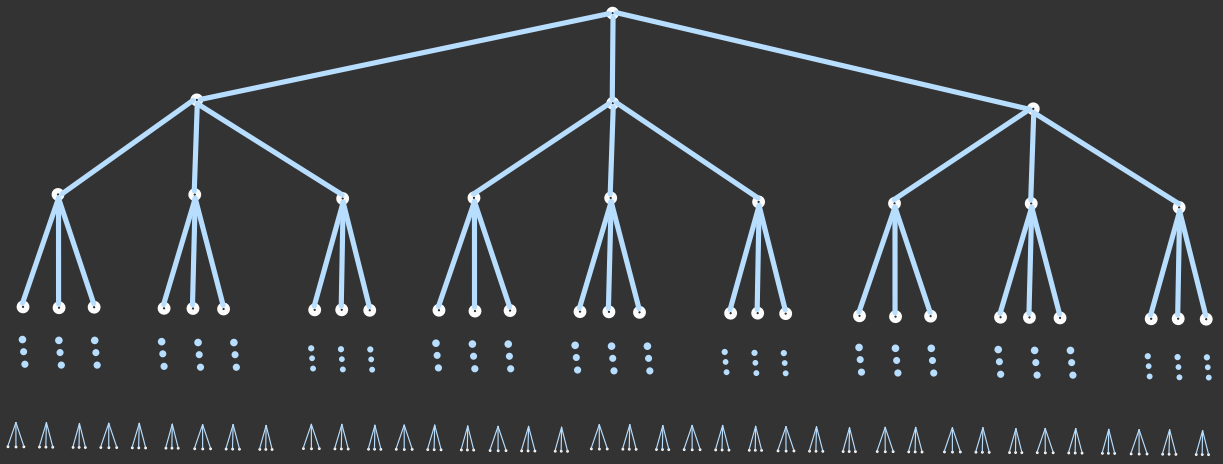
CSCI 373



depth	# nodes
0	
1	
2	your
3	answer
d	here

in a full tree with branching factor b

how many nodes are at depth d ?

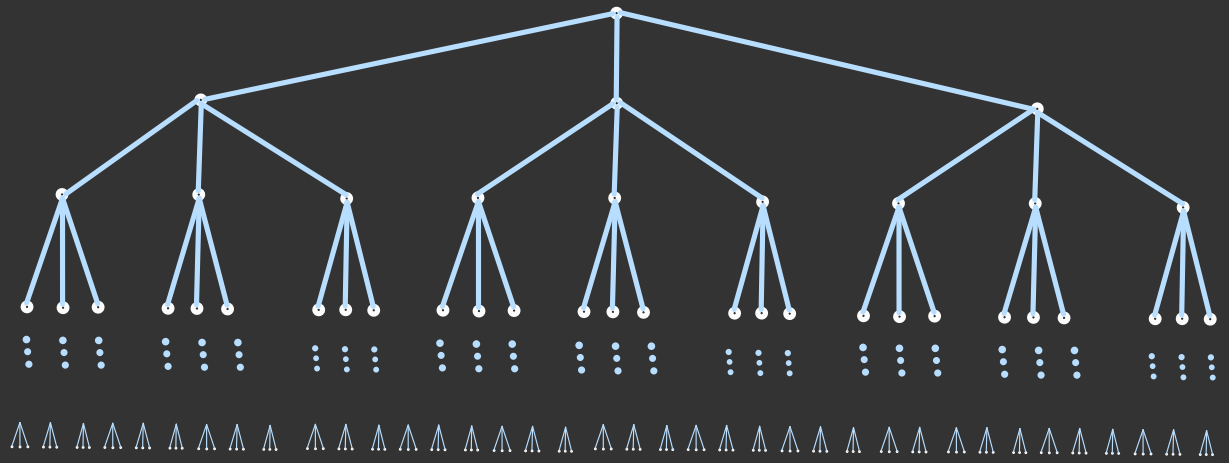


depth	# nodes	
0	1	b^0
1	3	b^1
2	9	b^2
3	27	b^3
d		b^d

in a full tree with branching factor b

how many nodes are at depth d ?

b^d



depth	# nodes	
0	1	b^0
1	3	b^1
2	9	b^2
3	27	b^3
d		$+ b^d$

in a full tree with branching factor b

how many nodes are there in total ?

$$\sum_{i=0}^{d} b^i$$

fun fact: if $b \geq 2$, then

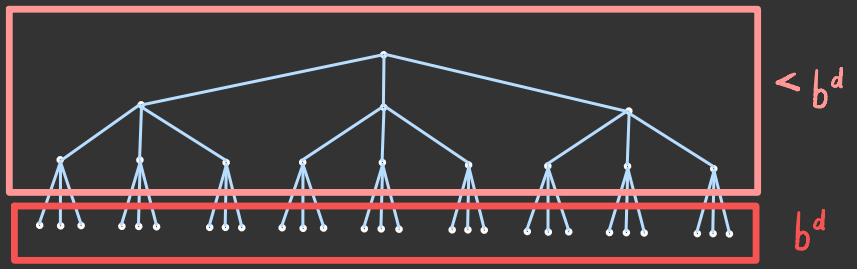
$$\sum_{i=0}^{d} b^i < 2b^d$$

given that we just observed there are b^d nodes at depth d of a full tree, what does this imply about the number of **internal nodes**?

your answer here

fun fact: if $b \geq 2$, then

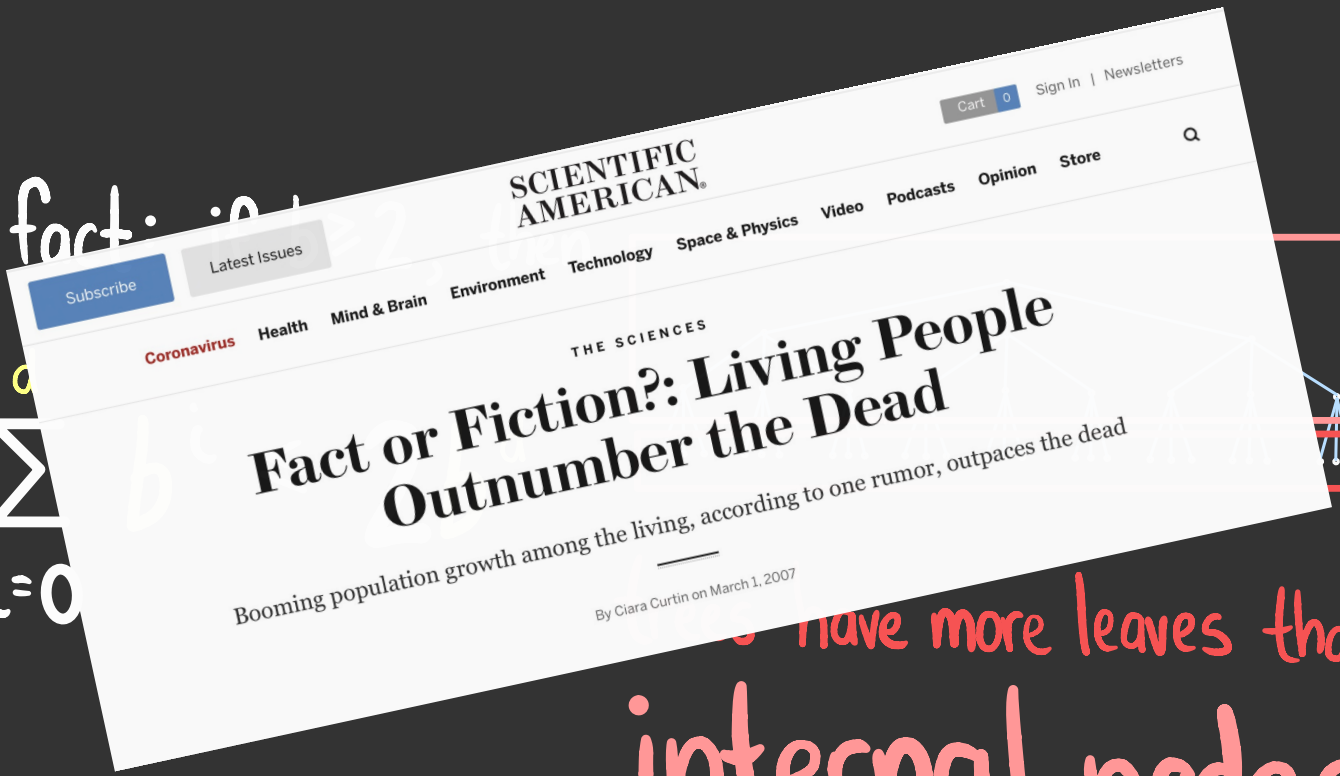
$$\sum_{i=0}^{d} b^i < 2b^d$$



trees have more leaves than internal nodes

fun fact:

$\sum_{i=0}$



$< b^d$

b^d

have more leaves than

internal nodes

Everybody feels they are right- some are confident, some are open. I'm definitely open to all of these possibilities, but science has it's say(irriducible complexity, creation- i.e. Einsteins clearly stated belief for existence, micro vs macro evolution). Personally that say has led me to be more inclined to the 6000 years theory (and this is ALL THEORY) but I remain open. Thus, if I had to choose today I'd say there are more alive today than ever before. But I'm (hopefully) always open for more info to bring more fact/truth

Jamie, Thorold Canada

No, there are definitely more people alive than dead. That is FACT!

Pippy Van't-Scmit, Kloofenhoffen Namibia

There are more dead people today than living.

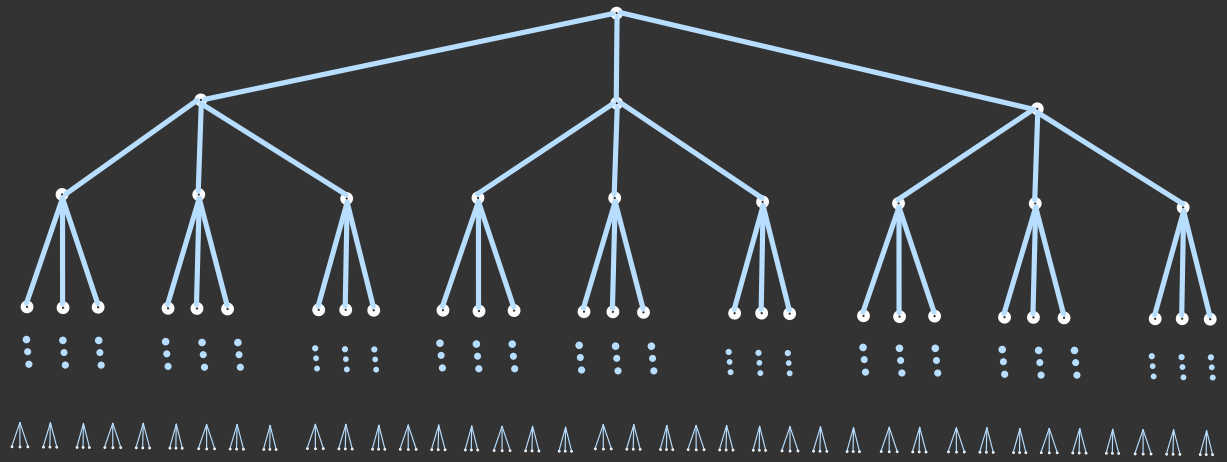
Abdulai Ibrahim, Tamale Ghana

6000 years? Really? I thought it was just some backward bible belt Americans who did not believe in evolution. Flat earth next maybe eh?

Barry Ley, Dubai UAE

Did some actually state the earth at 6000 years old? Why aren't they dead from stupidity? Ugh. Whatever the number is, there are a lot more brain dead people than alive.

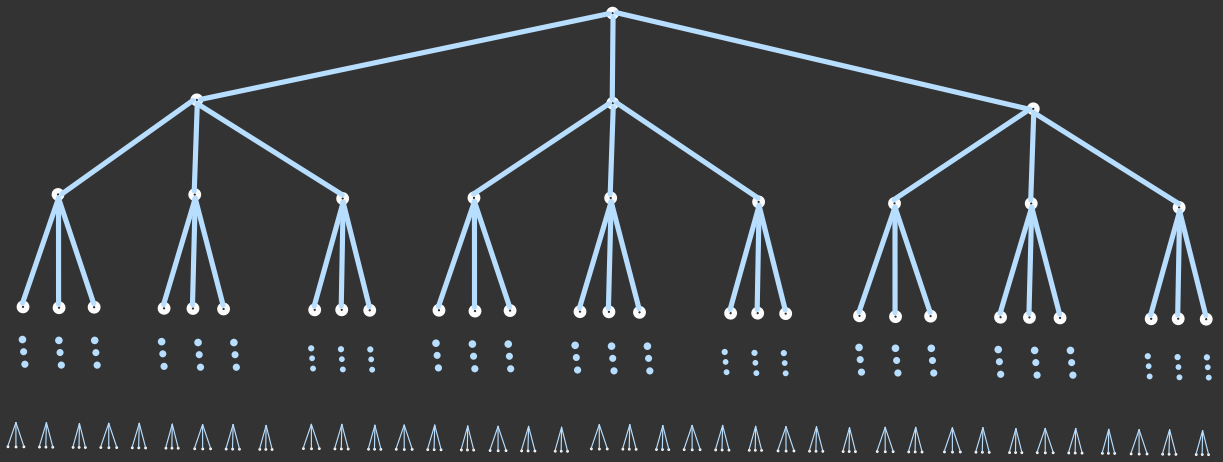
Tom, New York United States



in a full tree with branching factor b
how many nodes are there in total ?

$$\sum_{i=0}^{d} b^i < 2b^d = O(\boxed{?})$$

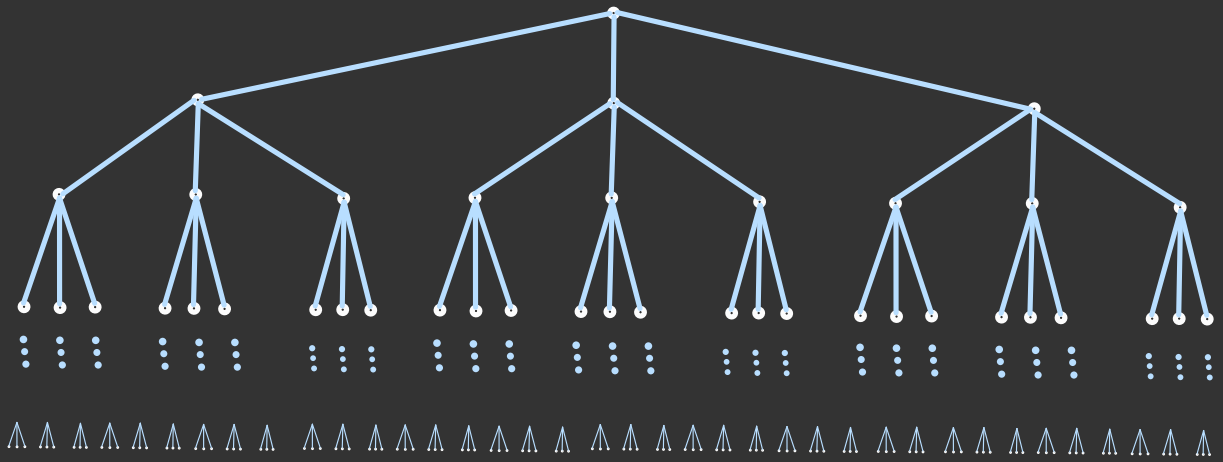
if $b \geq 2$

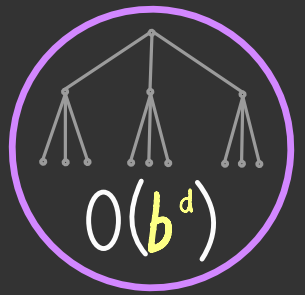


in a full tree with branching factor b
how many nodes are there in total?

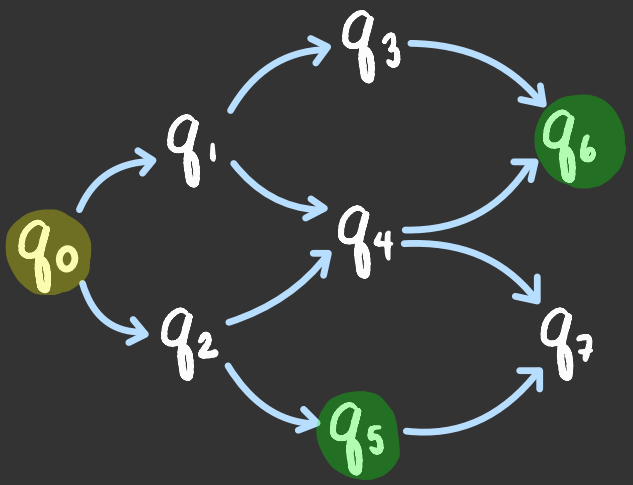
$$\sum_{i=0}^{d} b^i < 2b^d = O(b^d)$$

if $b \geq 2$

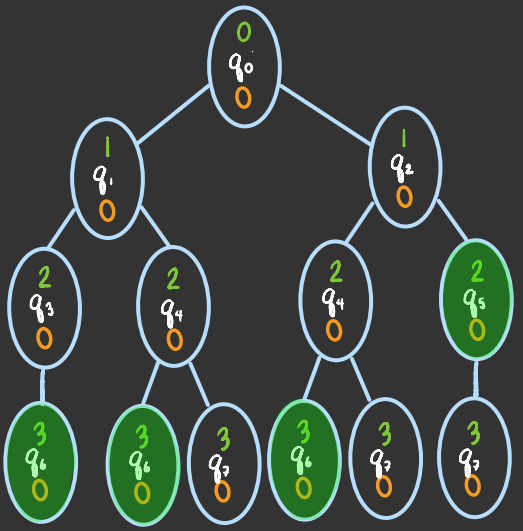




: a full tree with branching factor b and depth d has $O(b^d)$ nodes



state machine



search tree

d — **solution depth**

the length of the shortest search path that leads to a final state

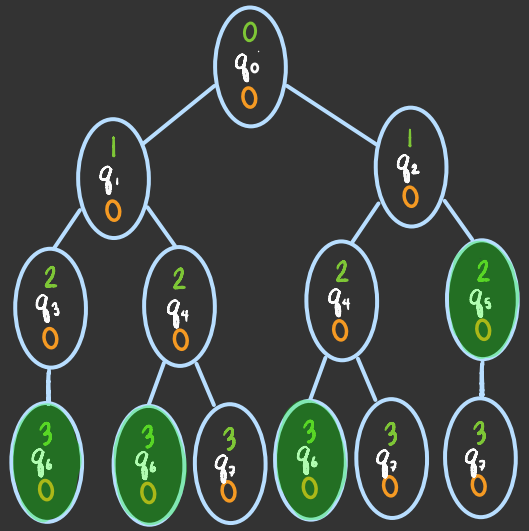
m — **maximum depth**

the length of the longest search path

b — **branching factor**

the maximum number of successors of a search node

what is the worst-case running time of bfs?



search tree

---

d — **solution depth**

the length of the shortest search path that leads to a final state

m — **maximum depth**

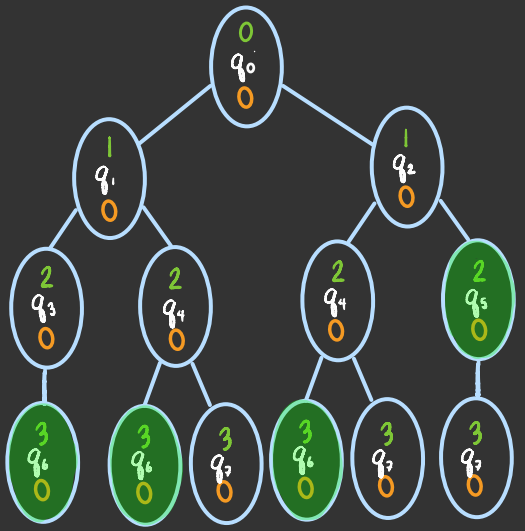
the length of the longest search path

b — **branching factor**

the maximum number of successors of a search node

recall this property of bfs:

if $j < k$, then every node at search depth j is visited before any node at search depth k



search tree

---

d — **solution depth**

the length of the shortest search path that leads to a final state

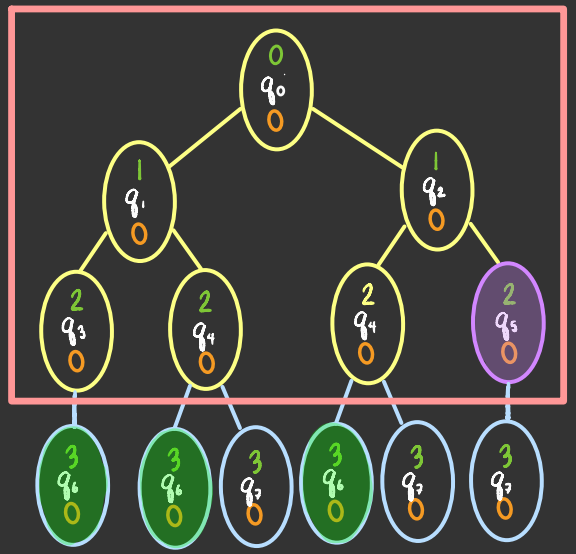
m — **maximum depth**

the length of the longest search path

b — **branching factor**

the maximum number of successors of a search node

so the worst-case scenario is that bfs visits all search nodes at solution depth d or lower

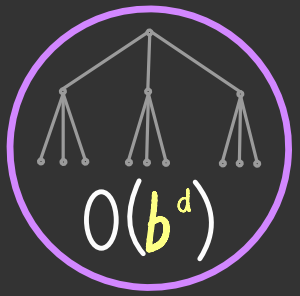


search tree

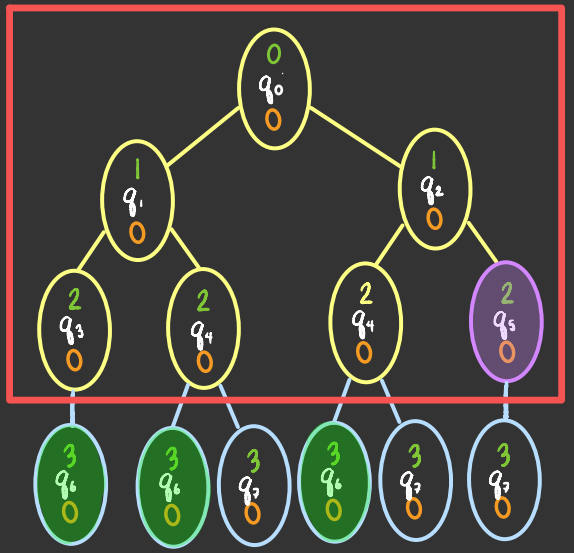
d — **solution depth**
the length of the shortest search path that leads to a final state

m — **maximum depth**
the length of the longest search path

b — **branching factor**
the maximum number of successors of a search node



so in the worst case, bfs visits $O(b^d)$ nodes



search tree

---

d — **solution depth**

the length of the shortest search path that leads to a final state

m — **maximum depth**

the length of the longest search path

b — **branching factor**

the maximum number of successors of a search node

what is the worst-case running time of dfs?

your answer here

d — **solution depth**
the length of the shortest search path that leads to a final state

m — **maximum depth**
the length of the longest search path

b — **branching factor**
the maximum number of successors of a search node

what is the worst-case running time of dfs?!

infinite

or

O(😭)

or

O(🚽)

---

d — **solution depth**

the length of the shortest search path that leads to a final state

m — **maximum depth**

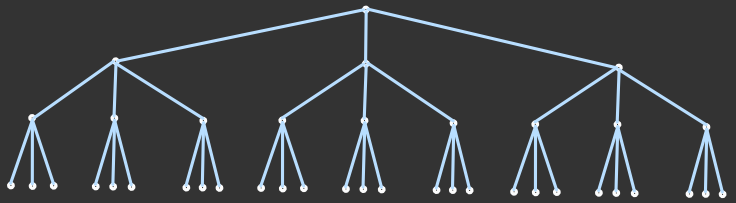
the length of the longest search path

b — **branching factor**

the maximum number of successors of a search node

what is the worst-case running time of dfs?

but what if we know that the search tree has finite depth?



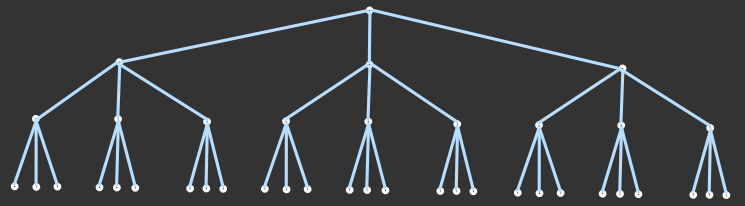
d — **solution depth**
the length of the shortest search path that leads to a final state

m — **maximum depth**
the length of the longest search path

b — **branching factor**
the maximum number of successors of a search node

what is the worst-case running time of dfs?

but what if we know that the search tree has finite depth?



there might not be a solution, or it be in the last place we look

d — **solution depth**
the length of the shortest search path that leads to a final state

m — **maximum depth**
the length of the longest search path

b — **branching factor**
the maximum number of successors of a search node

what is the worst-case running time of dfs?

so in the worst-case we need to visit all nodes in a tree of depth **m** and branching factor **b**, which is

$O(?)$

d — solution depth
the length of the shortest search path that leads to a final state

m — maximum depth
the length of the longest search path

b — branching factor
the maximum number of successors of a search node

what is the worst-case running time of

dfs?

so in the worst-case we need to visit all nodes in a tree of depth m and branching factor b, which is

$$O(b^m)$$

d — **solution depth**
the length of the shortest search path that leads to a final state

m — **maximum depth**
the length of the longest search path

b — **branching factor**
the maximum number of successors of a search node