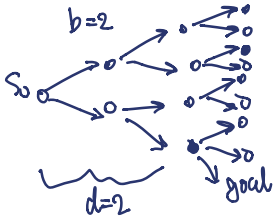
Assume our graph is ~~a DAG, a DTG is a tree~~ is a tree with branching factor b
"closest goal" is at depth d



depth, number of edges

tricky issues: ① multiple goals, at different depths
② tree can be infinite deep

Cost $(s \to t) = 1$, for all edges $s \to t$
Then PQ is a queue $\Rightarrow$ BFS

- Completeness ✓: Will find the goal if it exists.
- optimality ✓: Will find the shortest goal.
- Space complexity: $O(b^d)$ ↑ worst and (bad)

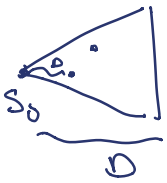
$f = O(g)$ if $f, f(n), g(n)$ are functions of n
$\exists C > 0$ |
$\exists N : \forall n \geq N \quad |f(n)| \leq C\,|g(n)|$

- time complexity: $O(1 + b + b^2 + b^3 \cdots b^d) = O(b^{d+1})$ (bad)

---

idea: replace $c + cost(s \to t)$ with a loop counter $\Rightarrow$ DFS

Incomplete:
assume tree finite depth D



optimality ✗

space: $O(d(b-1))$ very good.

time: $O(b^{d+1})$

---

better? Starting from DFS But:

For threshold $= 1, 2, 3, \ldots$
(depth cut off)
Do DFS from s_0, but cutoff at level threshold.

completeness ✓
optimal ✓
space: $O((b-1)\,d)$ ✓
time: