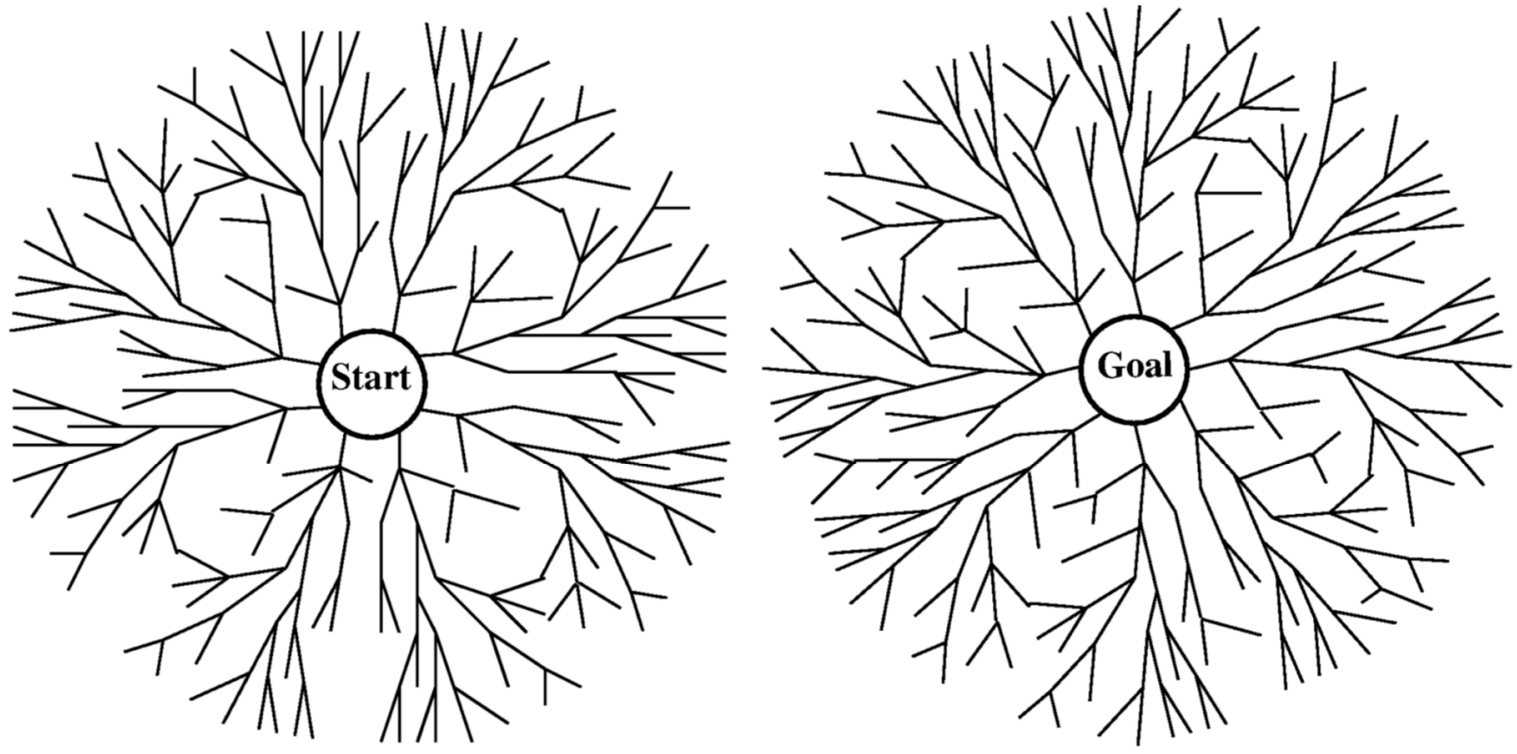
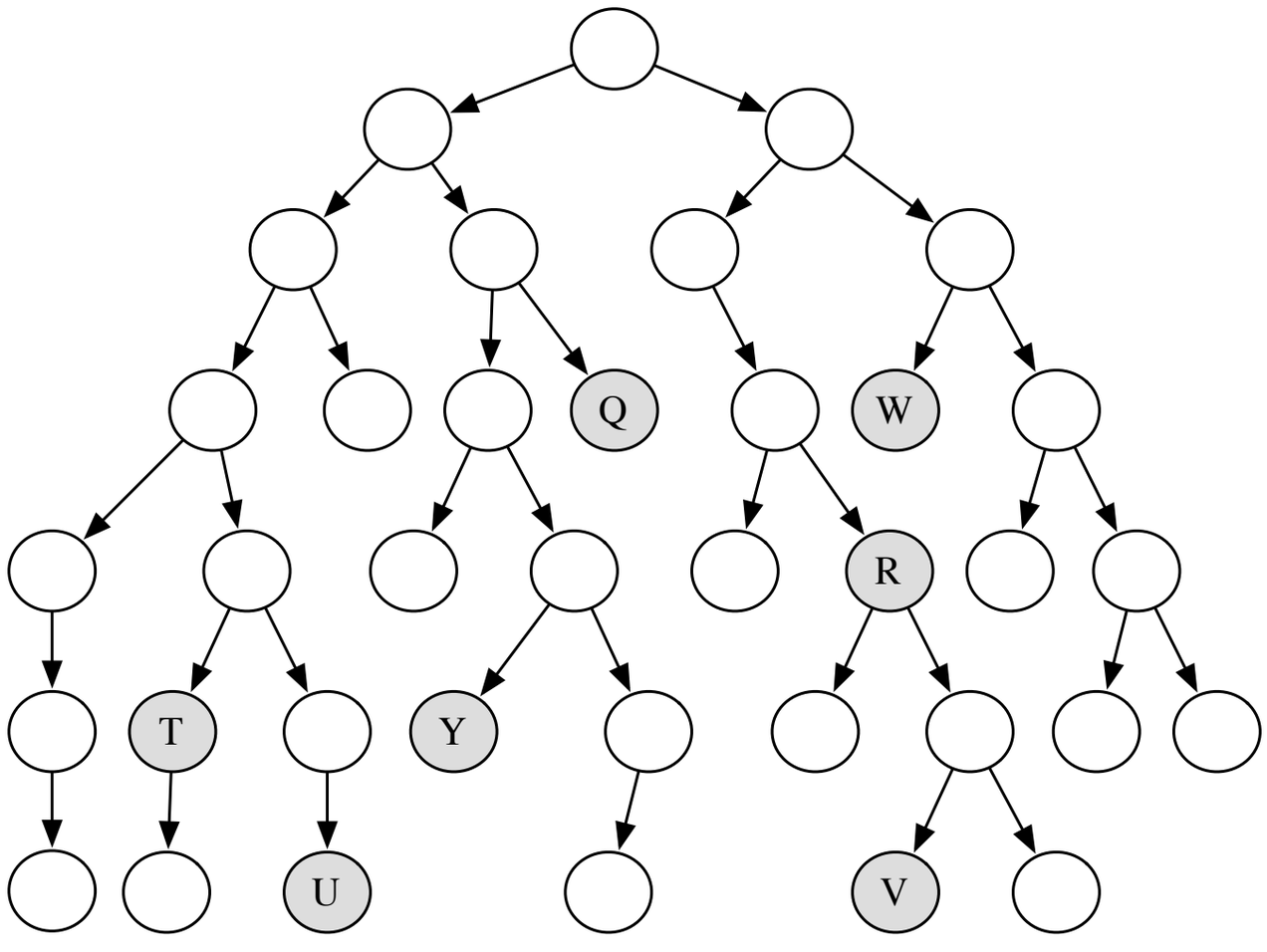
Bidirectional search

- Idea: search backward from the goal and forward from the start simultaneously.
- Can be used with BFS, LCFS, or A*
- This wins as $2b^{d/2} \ll b^d$. This can result in an exponential saving in time and space.



Bounded depth-first search

- A bounded depth-first search takes a bound (cost or depth) and does not expand paths that exceed the bound.
  - explores part of the search tree
  - uses space linear in the depth of the search.
- Which shaded goal will a depth-bounded search find first?



Iterative-deepening search

- Iterative-deepening search:
  - Start with a bound *b = 0*.
  - Do a bounded depth-first search with bound *b*
  - If a solution is found return that solution
  - Otherwise increment *b* and repeat.

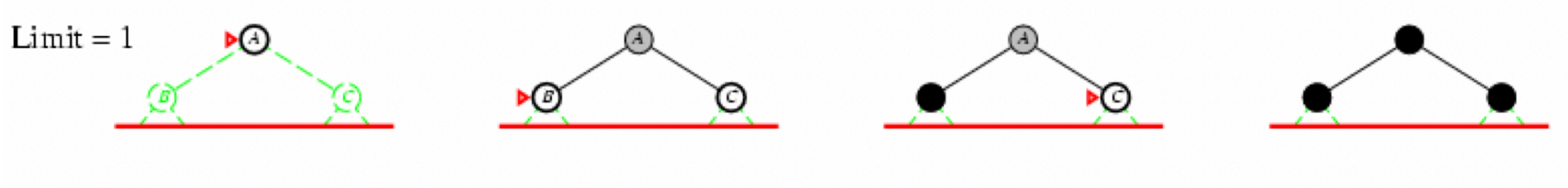
- This will find the same first solution as what other method?

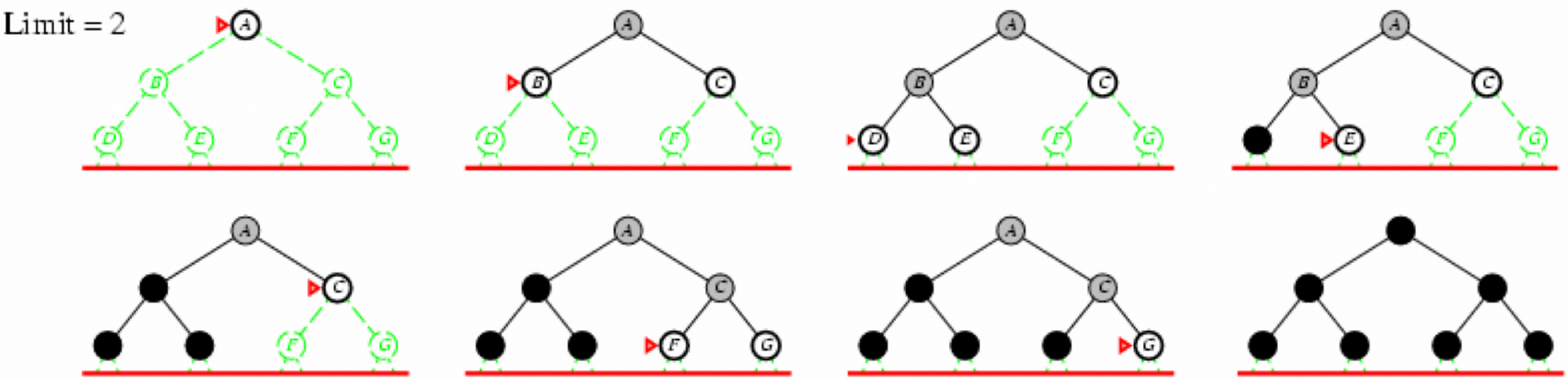
- How much space is used?

- What happens if there is no path to a goal?

- How wasteful is recomputing paths?

Iterative-deepening search: illustration





Iterative-deepening search: complexity

Complexity with solution at depth k & branching factor b:

level	breadth-first	iterative deepening	# nodes
1	1	k	b
2	1	$k-1$	b^2
...	...	...	...
$k-1$	1	2	b^{k-1}
k	1	1	b^k
total	$\geq b^k$	$\leq b^k \left(\frac{b}{b-1}\right)^2$	