Generic Search Algorithm

Breadth-First

Procedure *Search*(G, S, *goal*)

Inputs: G: graph with nodes N and arcs A

S: set of start nodes

goal: Boolean function of nodes

Output: path from s to g s.t. $s \in S$ and *goal*(g) or null if no solution paths are found

Frontier $\leftarrow \{(s) \mid s \in S\}$ // a set of paths

while *Frontier* is not empty

dequeue

~~remove~~ a path $p = (s, \ldots, t)$ from *Frontier*

if *goal*(t) then return p

for each arc from t to n

enqueue

~~insert~~ $(s, \ldots, t, n)$ into *Frontier*

return null

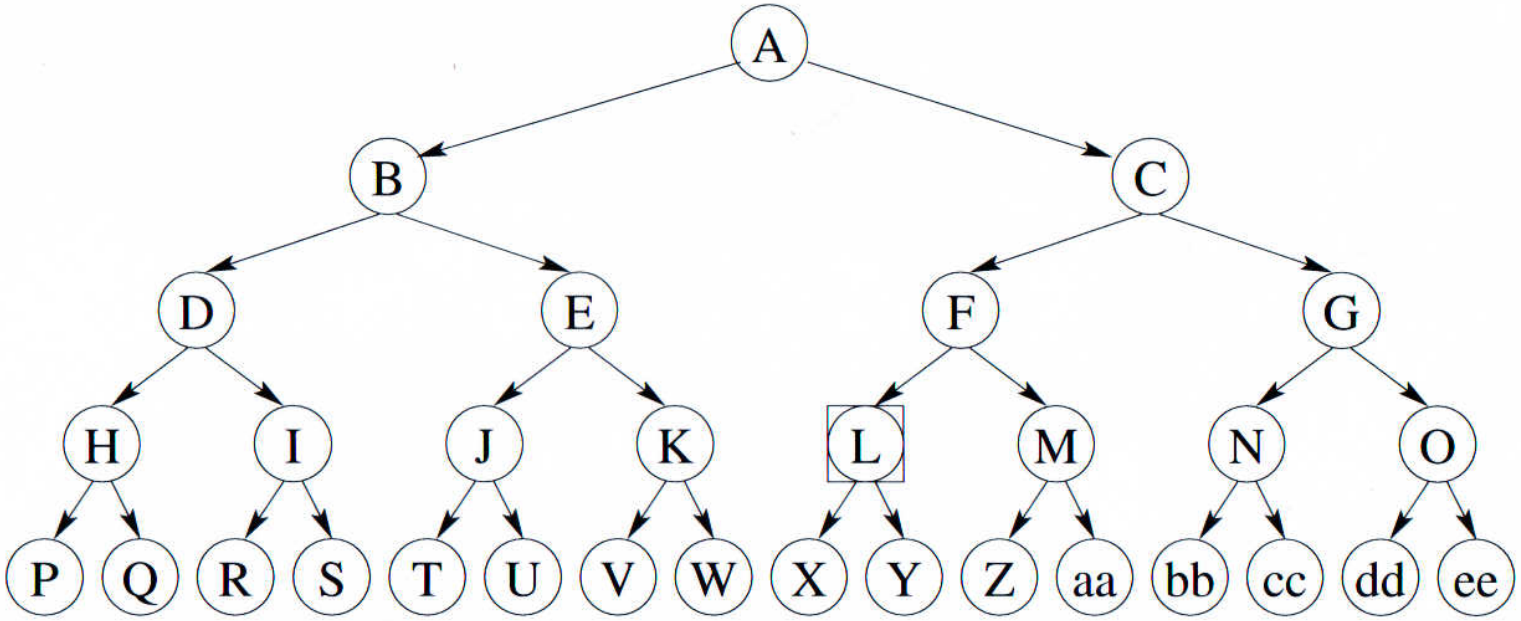
This is a queue

first-in first-out

multiple path pruning

don't do this if t has already been processed

Uninformed Example



IDS

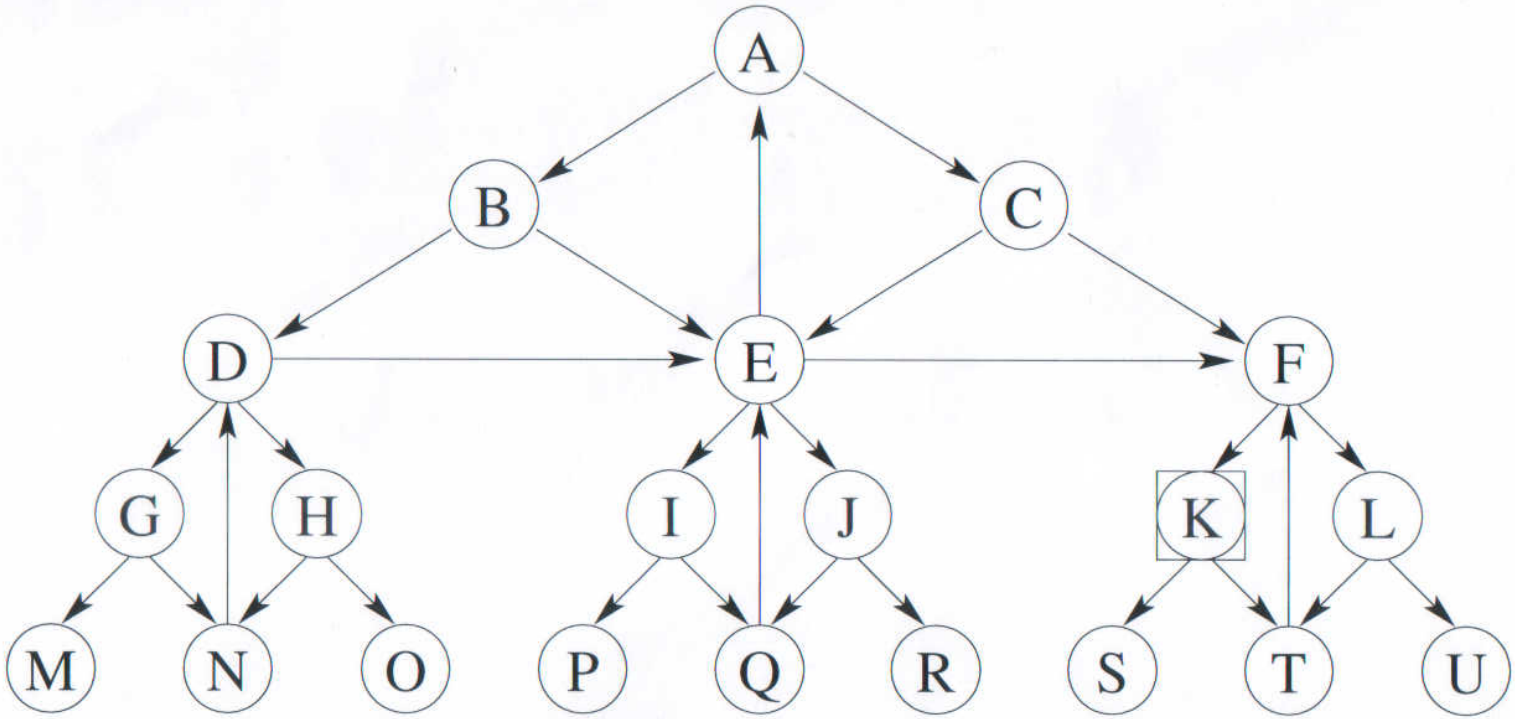
bound = 0 A

bound = 1 A B C

bound = 2 A B D E C F G

bound = 3 A B D H I E J K C F L

Uninformed Example 2

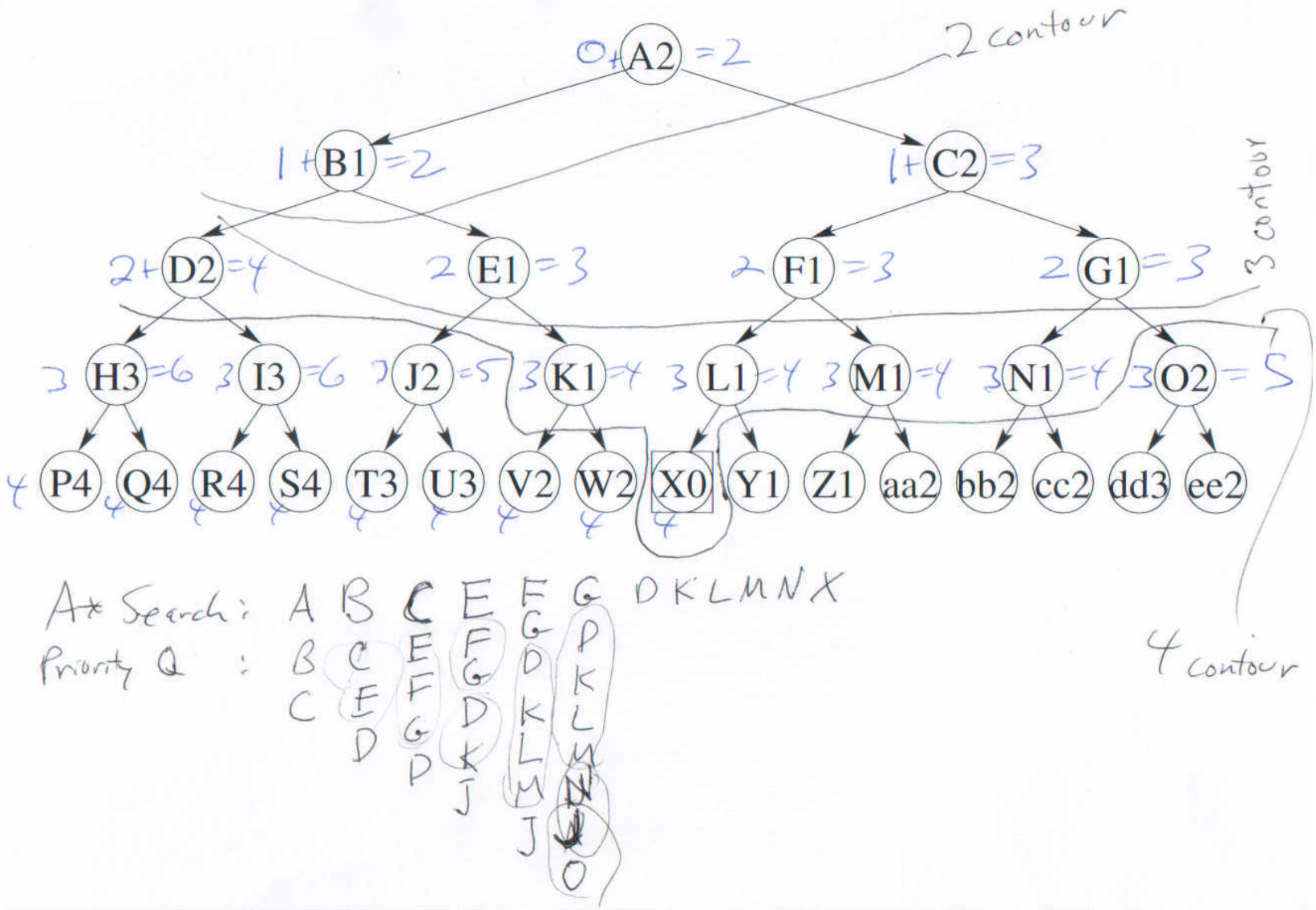


DFS with cycle-checking: A, B, D, G, M, N, H, N, O, E, I, P, Q, J, Q, R, F, K
end with (A, B, D, E, F, K) path

ID: A, A B C, A B D E C E F, A B D G H E E I J F C E I J F F K

BFS: A B C D E F G H I J K L
?
assumes multiple-path pruning

Heuristics Example



Experiment Avoiding Reverse Moves

h_1 = misplaced tiles h_2 = Manhattan distance

	States Visited (Effective BF)		
d	ID	IDA*(h_1)	IDA*(h_2)
4	52 (2.35)	10 (1.35)	7 (1.17)
8	569 (2.03)	42 (1.36)	14 (1.11)
12	5357 (1.92)	315 (1.47)	45 (1.19)
16	47271 (1.87)	2410 (1.52)	226 (1.28)
20		17646 (1.55)	764 (1.29)

$10 = x^0 + x^1 + x^2 + x^3 + x^4$ solve for x

$x = 1.35$ is closer to 10

$17646 = x^0 + x^1 + x^2 + \cdots + x^{20}$ solve for x