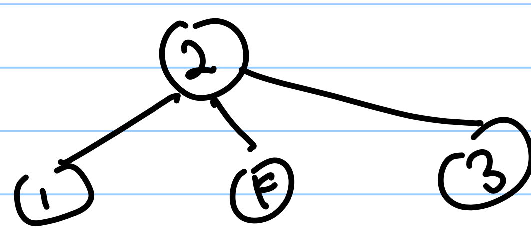
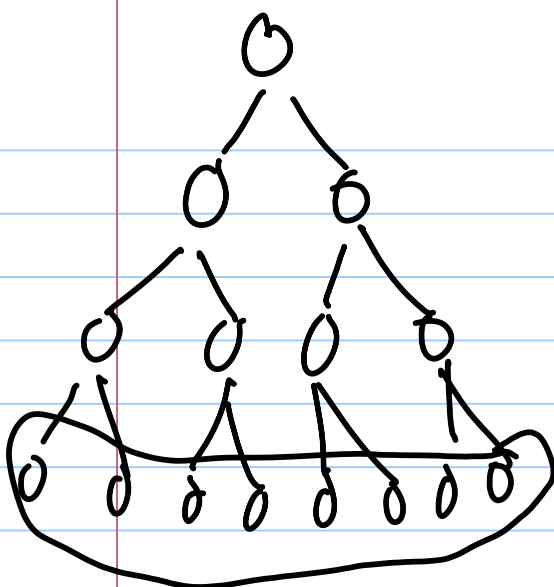
Note Title 2/11/2008

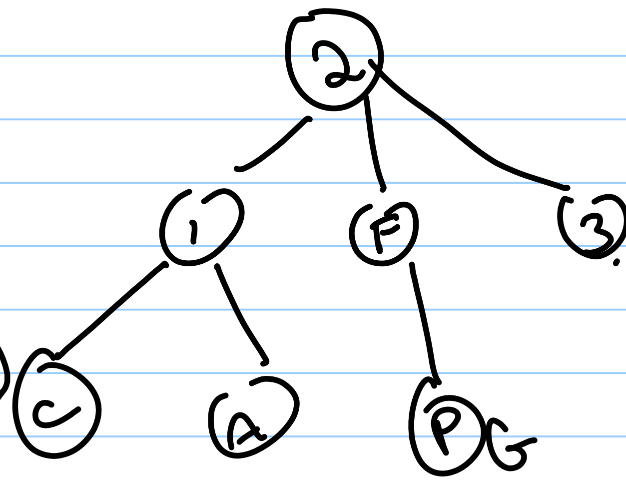
Uninformed Search (Ch 3) - Informed Search (chp 4)

Depth Limited Search - modification of depth first - set a limit on the depth to which you search

Iterative deepening try each successive depth in turn!
Combines breadth first and depth first





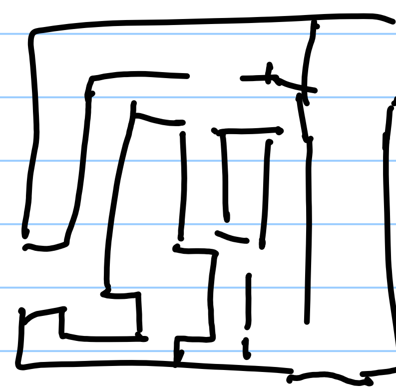


States may be repeated

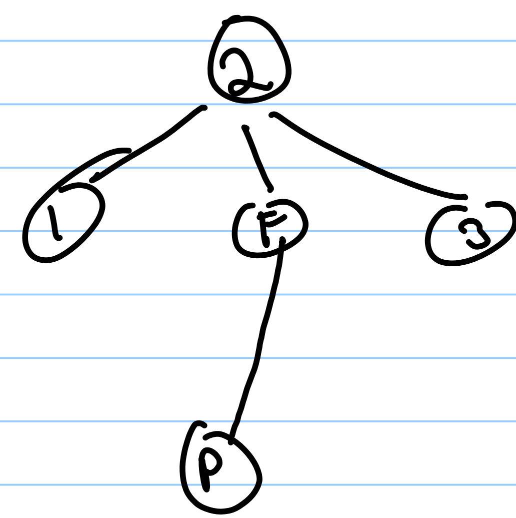
This is not so bad since an exponential problem has most nodes in the bottom level

Bidirectional Search

search forward from initial state and backward from goal state. Solution is where they meet



$O(b^{d/2})$ since each search has to go only halfway



Downside - to search backwards we need a predecessor function. Sometimes calculating predecessors can be difficult

We need to know the goal

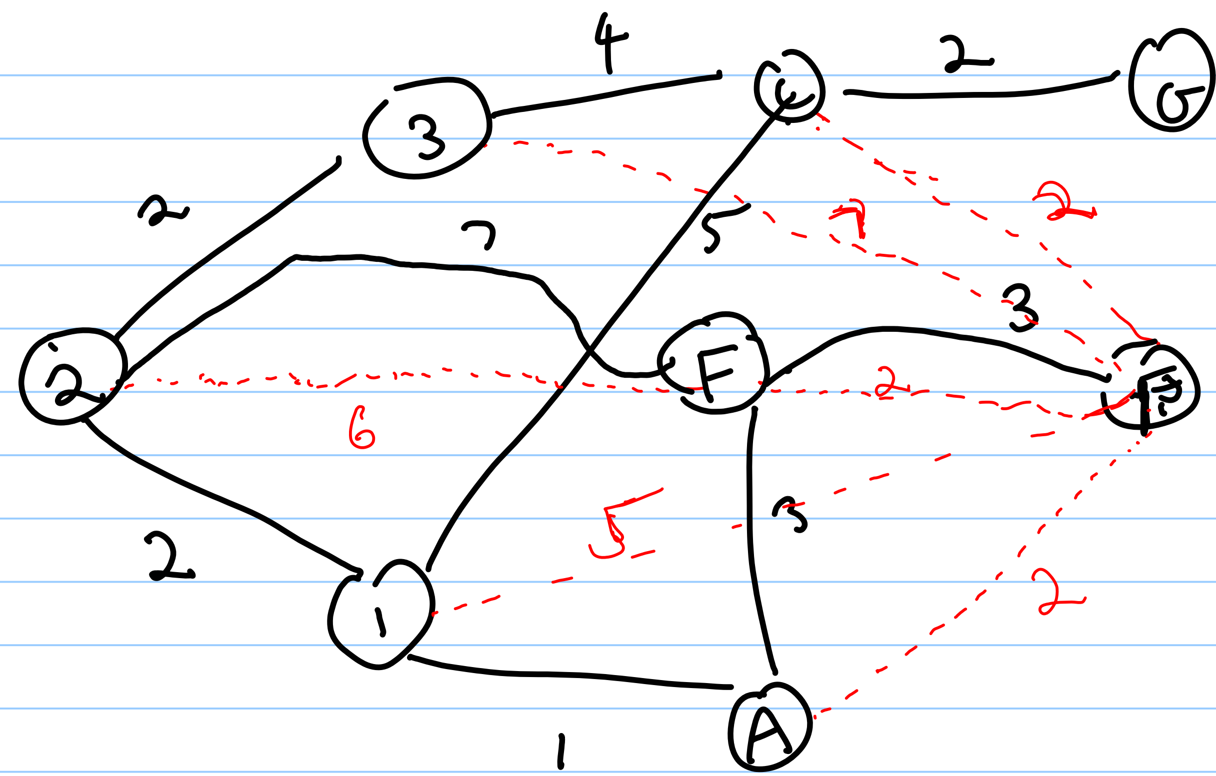
Efficiency - we need some efficient way to search the fringe

Which algorithm to search each half of the search space

Informed Search

yeah!!

Use knowledge to guide our path toward the goal



General methodology - Best first search

When deciding which path or terminal node to expand use knowledge obtained from an evaluation function to pick the one with the best evaluation value.

Heuristic function - helps us to estimate which node is closest to the goal.

Heuristic - a method that uses knowledge to shorten search

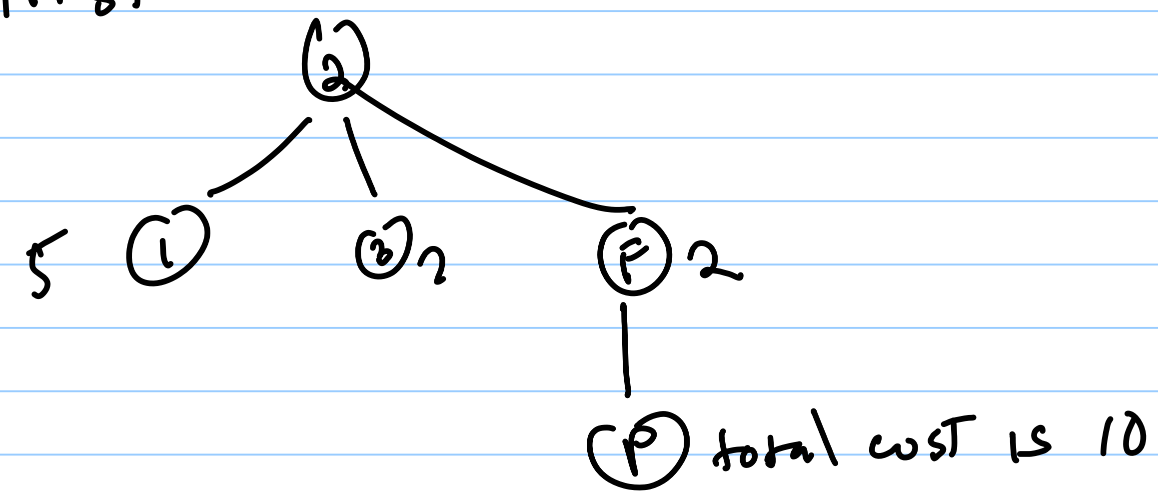
Gives estimated cost of cheapest path from node n to the goal state

Greedy best first search

$h(n)$

Straight line distance

Expand the node with the closest SLD to P first



Not Optimal

Finds a solution quict

[illegible]

Can be susceptible to false starts since it may expand a path that does not lead to a goal but its nodes are close to the goal.

If we don't test for repeated states can lead to infinite search

Greedy bF resembles Depth First

time complexity $O(b^m)$ m = maximum depth

memory $O(b^m)$

Uniform cost search - expands shortest path thus far $g(n)$

Greedy BFS expand path with shortest estimate to goal $h(n)$

Best $= g(n) + h(n)$

A* search

$f(n) = g(n) + h(n)$

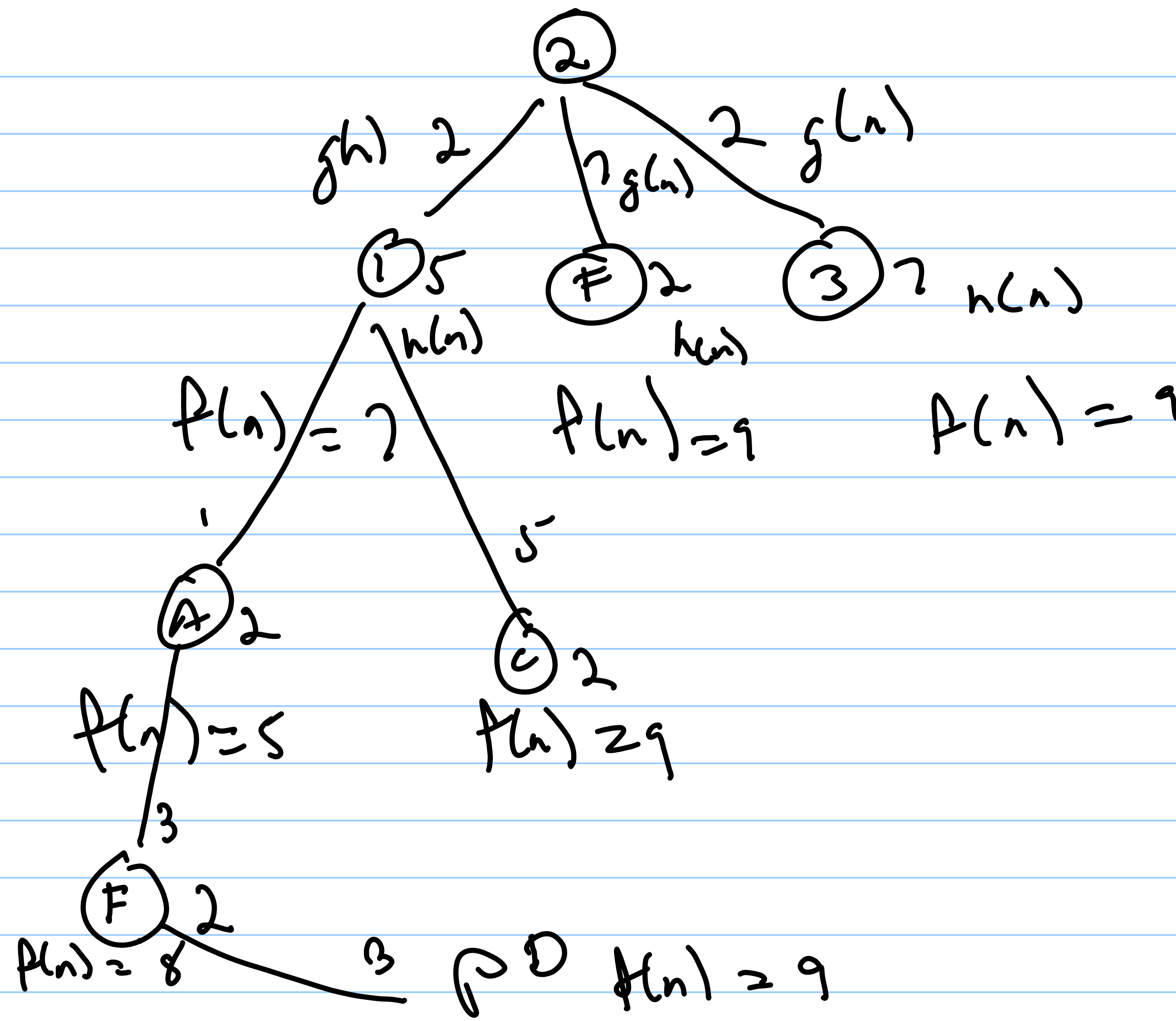
$g(n)$: distance thus far

$h(n)$: heuristic (SLD)

$f(n)$ - estimated cost of the cheapest solution through node n

Complete & optimal as long as h (heuristic) doesn't overestimate the cost of reaching the goal (admissable heuristics)

If h is admissable then $f(n)$ never overestimates the actual cost of the best solution through n.


2
g(n) 2
7 g(n)
2 g(n)
1 5
F 2
3 7
h(n)
h(n)
h(n)
f(n) = 7
f(n) = 9
f(n) = 9
1
5
A 2
C 2
f(n) = 5
f(n) = 9
3
F 2
f(n) = 8
3
D
f(n) = 9

A^* is optimal