Search

Lecture 2
January 30, 2007

Discussion of assignments and programming

Problem Solvable Using Search

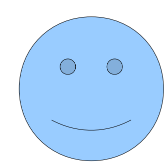
- Assumptions about Problems
  - Static
  - Observable
  - Discrete
  - Deterministic
  - (Often) Markovian
- Definition of Problems
  - Initial state
  - Successor Function
  - Path Cost
  - Goal

Problems that are amenable to Search

- The 8 puzzle
  - Initial: some organization of tiles
  - Goal: Some other organization of tiles
  - Successor: moving around the blank
  - Path cost: just 1
- 8 Queens
- Route Finding
- Traveling Salesman
- Bin Packing

State Spaces

- Tic-tac-toe
  - 3^9=19683, but after symmetry, etc = 765
- N Puzzle
  - 8 -- 9!/2=181,440
  - 15 -- 16!/2=1,300,000,000,000
  - 24 – 25!/2=10^25
- N-Queens
  - 1.8*10^14
- Traveling Salesman
  - N!
  - Can be solved in 2^N (2^N << N!)



Searching the State Space

- General Algorithm

```
fringe <- initial State
A: s <- first state from fringe
if s==GOAL then stop
p <- successors of s
fringe <- fringe union p
if fringe empty then stop
goto A
```

- (fringe union p) vs (p union fringe)

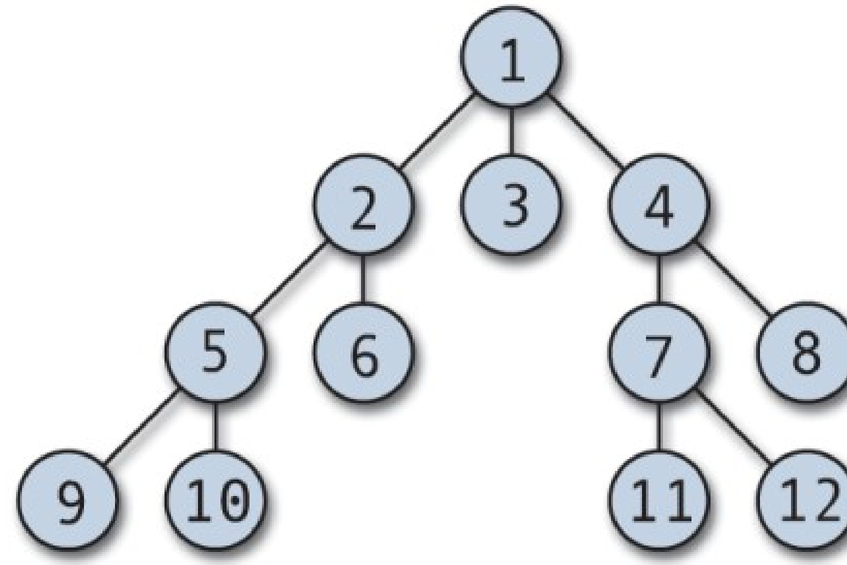
Evaluating Search

- Completeness
- Optimality
- Time
- Space
- Cost

- Branching Factor

Breadth First Search

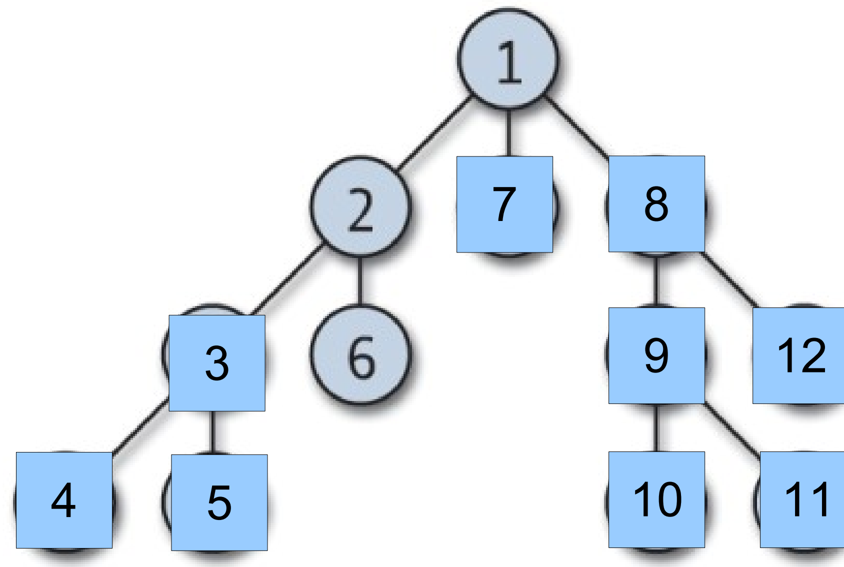
- Use the search algorithm with
  - fringe<- fringe union p
- Time & Space
  - O(V+E)
- Finds Optimal Solution
  - Yes if cost is a non-decreasing function of depth



http://www.cs.duke.edu/csed/jawaa2/examples/BFS.html

Depth First Search

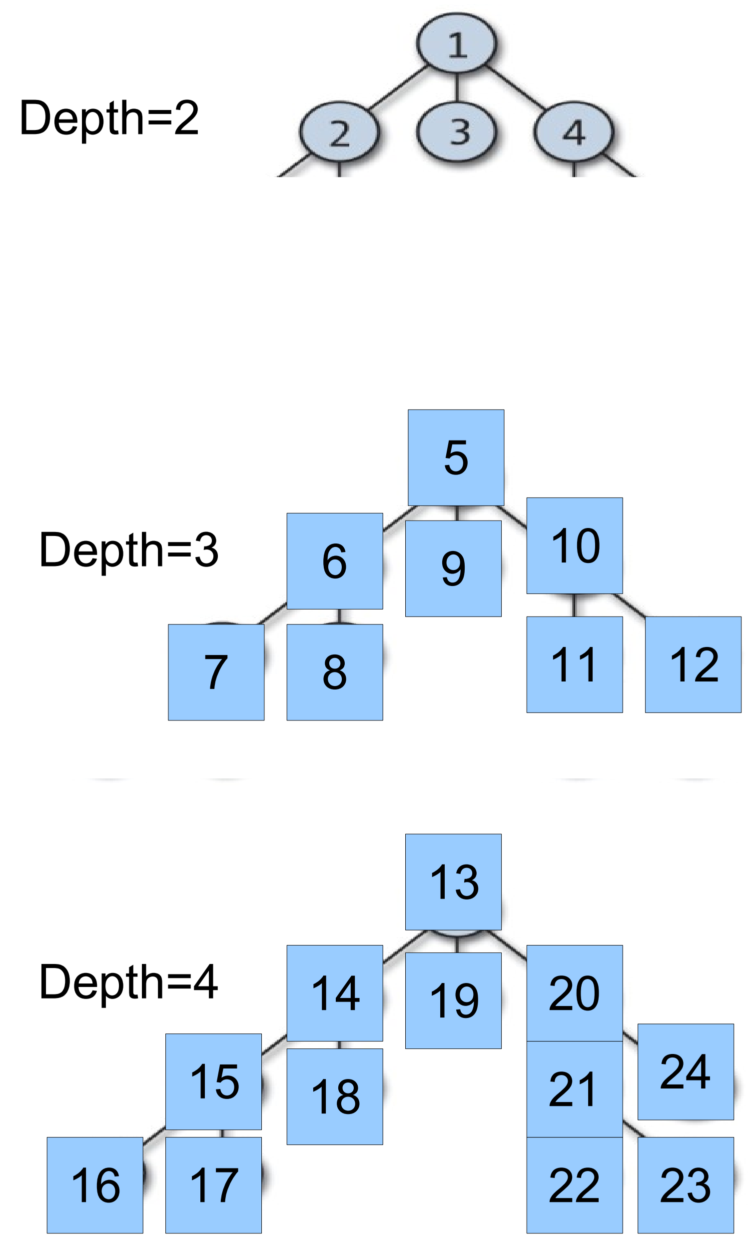
- Use the search algorithm with
  - fringe<- p union fringe
- Time
  - O(V+E), same as BFS
- Space
  - Better than BFS
- Finds Optimal Solution
  - Yes or No



http://www.rci.rutgers.edu/~cfs/472_html/AI_SEARCH/SearchAnimations.html

Iterative Deepening DFS

- Gets you best of DFS and BFS
- In a balanced tree time is at worst double
- Idea DFS to depth=1 then 2 then 3, ...



Other Uninfomed Searches

- Uniform cost
  - Applies BFS to links with transit cost
- Depth Limited
  - DFS but only so deep
- Bidirectional
  - BFS starting at beginning and end

Informed Search

- Key idea – use a guess to guide the selection of the next move.
  - 8-puzzle – guess might be number matching the goal
  - Navigation – straight line distance from goal
- “Best first” search
  - Expand the node that is closest to the goal.
    - As opposed to BFS or DFS
  - “Greedy”

A* Search

- Minimize the total cost of the solution
- $F'(n) = g(n) + h'(n)$
  - $F(n)$ == cost of solution going through node n
  - $g(n)$ == cost to get where you are (node n)
  - $h(n)$ == cost to get from node n to goal
  - ' indicates an estimate
- Admissable
  - h is admissable if $h'(n) < h(n)$
- If h is admissable then A* will find optimal solution

Learning & A* search

- Are there opportunities?

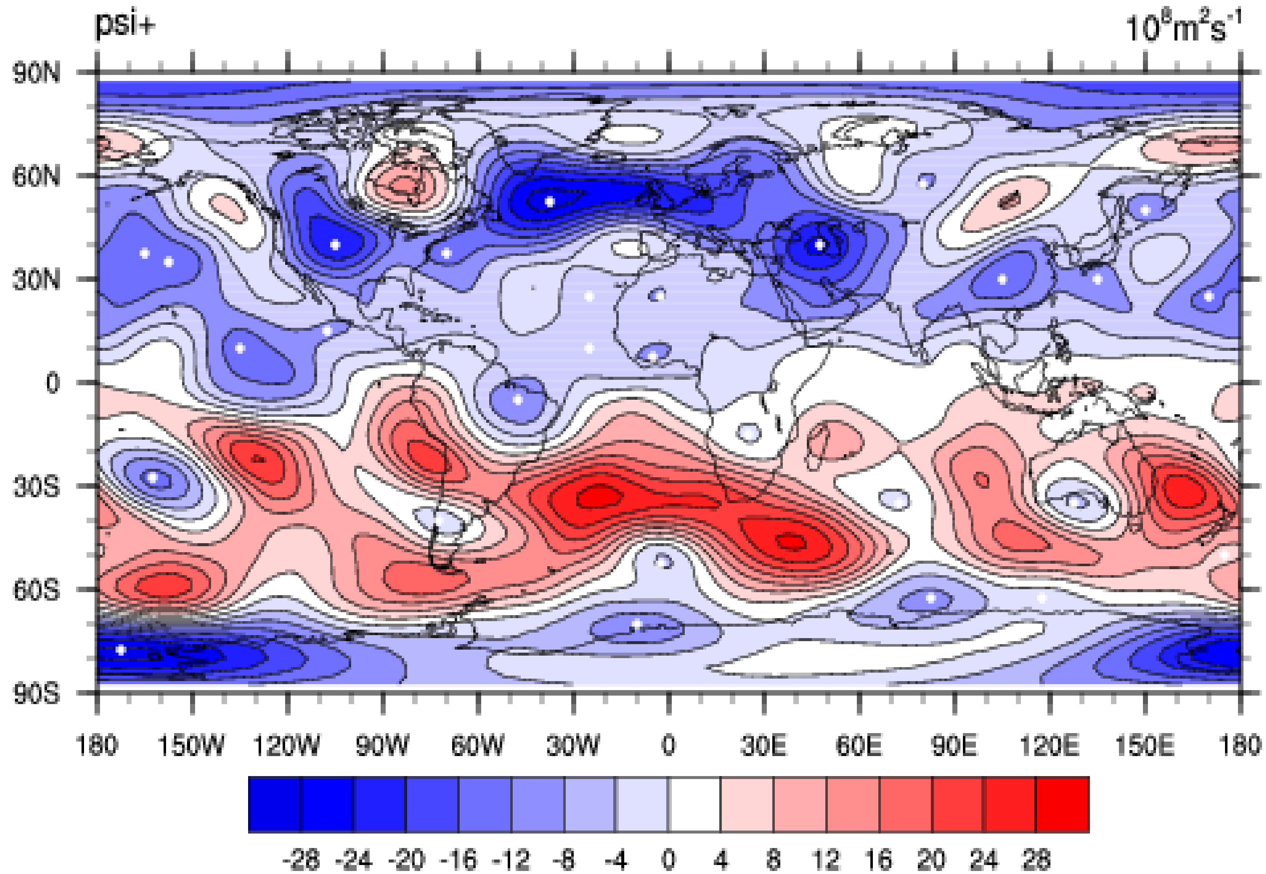
- What info do you need?
  - What is cost of keeping this info?

Hill Climbing

- Suppose you are looking for the highest mountain. One approach is to start walking up hill. At every step go up in the steepest direction.
- Problems?
- How is this like A* search?

Hill Climbing (continued)

- Local maxima (minima)
- Flat spots
- Global maxima
- Ridges
- Saddle points



- Neural networks, decision trees, ...

Hill Climbing -- “fixing”

- Random Restarts
  - Start in a different place, end up at a different high point
- Beam Search
  - Start at n random points. Find all successors of that set. Call these N'. Eliminate from N' all but the n with best h'(). Repeat.
- Simulated Annealing
  - Every once in a while, give everything a good shake, but shake a little less every time you shake.

AI Vocabulary

- Breadth First Search
- Depth First Search
- Branching Factor
- Best First Search, A*
- Open List, Closed List, fringe
- Hill Climbing
  - Flat spots, ridges, plateaus
  - Simulated Annealing, Random Restarts, Beam Search
- Greedy Functions
- Heuristics
  - admissible