Emmanuel Yeng
Wing Wong
Ron Peled
3/3/2004

Practice Exam

#1) **States:** any arrangement of 2 red tiles, 3 white tiles, and 3 black tiles on a 3×3 board. (one slot is empty).

Initial State: any state from the possible set of states.

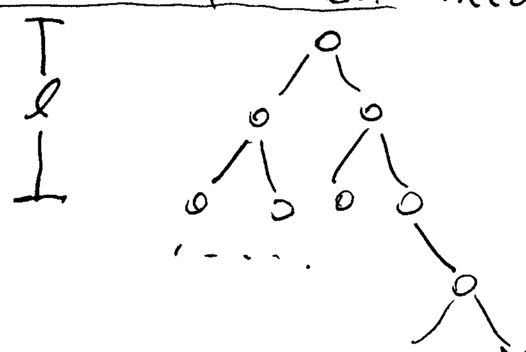
Successor function: Legal states that result from trying the four actions: blank moves Left, Right, Up, down.

Path Cost: each step costs 1. path cost = # of steps in the path from initial state.

2) **breadth first search:** complete.
since we are working with a finite set of states problems, and breadth first search covers systematically all nodes possible in the tree. ⇒ complete.

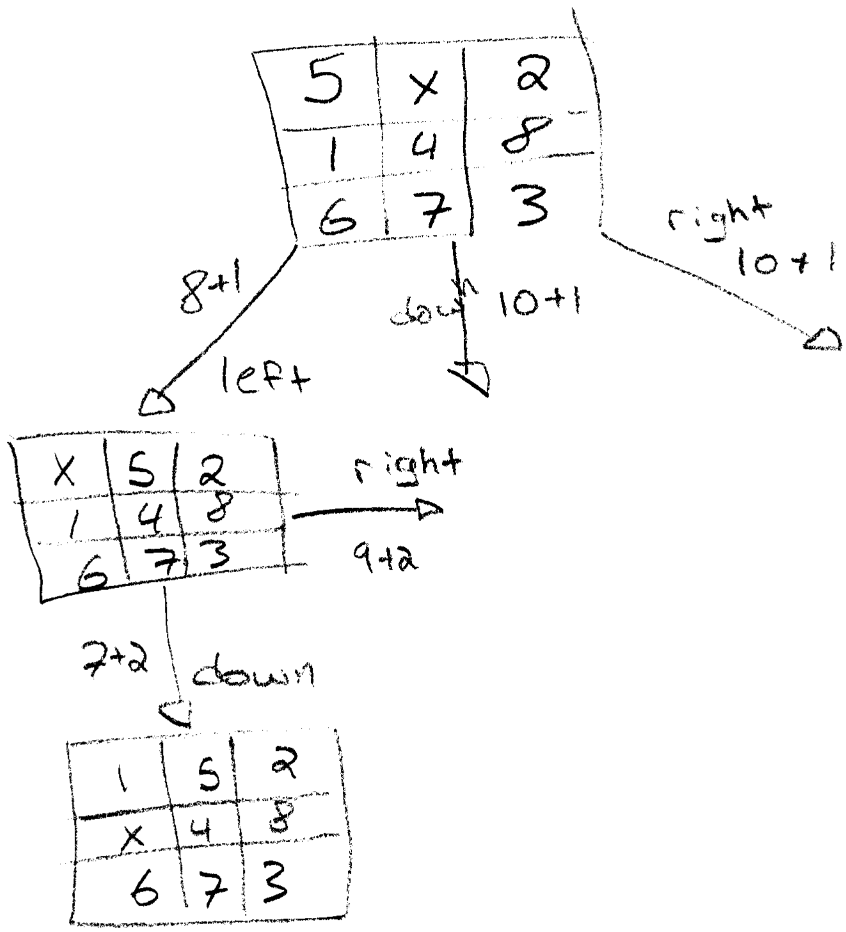
depth first search: not complete.
e.g. if left subtree is unbounded and contains no solution.

depth limited search: incomplete.



3. Step 1: expand a
Step 2: expand nodes d≤2, d expanded
Step 3: expand nodes d≤4, b & c expanded.

4.



(5) Simulated Annealing - an algorithm to find an optimal solution.

<u>Algorithm</u>

From a state choose neighbor at random
if neighbor is more optimal → switch
else switch with probability $e^{\frac{-K(\Delta E)}{\tau}}$

K - fudge factor

ΔE - how much worse is neighbor than current state

τ - Temperature

Lower Temperature after doing above several times, repeat the algorithm until answer is consistant.

<u>Important</u>: Knowing when to switch Temperature according to a schedule.

6. $f1(n1) = 10$ $\quad$ $f1(n2) = 7$

$f2(n1) = 7$ $\quad$ $f2(n2) = 10$

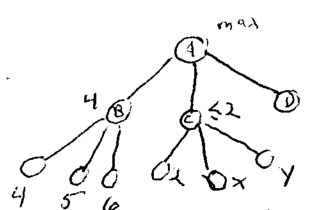
$sol = 15$ $\quad$ $sol = 20$

$$\boxed{c_1 f_1(n) + c_2 f_2(n) = h(n)}$$

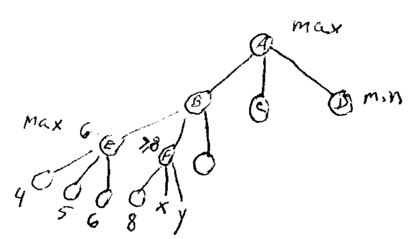
$$\begin{array}{l} (10c_1 + 7c_2 = 15)\,10 \\ (7c_1 + 10c_2 = 20)\,{-7} \end{array} \Rightarrow \begin{array}{rl} 100c_1 + 70 &= 150 \\ -49c_1 - 70 &= -140 \\ \hline 51c_1 &= 10 \end{array} \Rightarrow c_1 = \frac{10}{51} = .196$$

$$c_2 = 1.863$$

7)



α prune since value of node C will be ≤ 2



β prune since value of node F will be ≥ 8

~~max will always choose~~

8. Let x_0, x_1, x_2, x_3 represent the state of the light switches

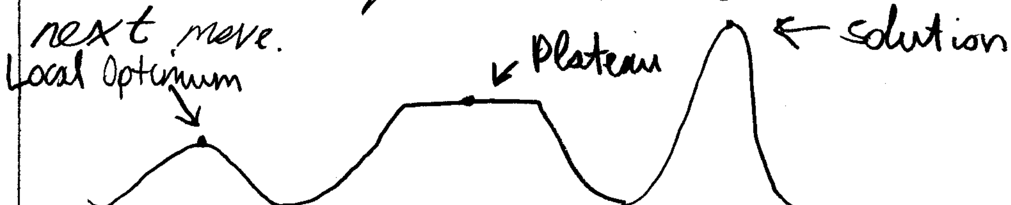
$$\neg(x_0 \wedge x_1) \wedge \neg(x_0 \wedge x_1 \wedge x_2 \wedge x_3)$$

Also, since $(x_0 \wedge x_1 \wedge x_2 \wedge x_3)$ cannot be true without $(x_0 \wedge x_1)$ being true,

$$\neg(x_0 \wedge x_1)$$

expresses the same equation.

9. Hill-climbing can get stuck at a local optimum where it would terminate prematurely, or at a plateau where there is no next move.



Vincent Ohm
Marvin Gouw

10 IDA* Iterative Deepening A*

Instead of having the cost value based on height, we base it on a threshold value. First step would be to expand all nodes whose heuristic weight is less than the threshold. Then if we haven't found our goal, we increase the threshold (if ε is threshold, next iterative step might be 2ε as threshold) Delete all nodes from previous step and expand nodes again as long as the heuristic weigh t is less than the current threshold value. Iterate the thresholds and continue in steps.

SMA* - Simple Memory-bound A*

SMA works by expanding all nodes until memory is filled up. At this point we drop the node with the worse calculated heuristic weight, We store the weight of this node in its parent in order to conserve the space of having that node in memory. Since it is the worse node, a simple weight can be stored - we are not probably going to expand it any time soon, Expand the best node. If node values are equivalent, expand the newest node and drop the oldest node (ties)