The Cognitive Side of Problem-Solving; and Game-Playing 1
CSCI 5582, Fall 2007
Problem-Solving: The Cognitive Side

• Do we actually use (say) depth-first search? Or breadth-first search? Or A*?
• Is the problem-space formalism useful, or natural?
• How do we solve problems, anyhow?
Hanging from the ceiling are two lengths of string, each six feet long. The strings are ten feet apart. As the room is ten feet high, the lower ends of the strings hang four feet above the floor.

On the floor is a Swiss Army knife, a firecracker, a small glass of water, a twenty-five pound block of ice in an aluminum pan, and a friendly Labrador retriever.

Your job is to tie the two ends of the two strings together.

-- based on Poundstone, "Labyrinths of Reason"
"A man gets an unsigned letter telling him to go to the local graveyard at midnight. He does not generally pay attention to such things, but complies out of curiosity. It is a deathly still night, lighted by a thin crescent moon. The man stations himself in front of his family's ancestral crypt. The man is about to leave when he hears scraping footsteps. He yells out, but no one answers. The next morning, the caretaker finds the man dead in front of the crypt, a hideous grin on his face.

Did the man vote for Teddy Roosevelt in the 1904 U.S. presidential election?"

-- Poundstone
All astronomers are stamp collectors.

No baseball players are astronomers.

∴ ???
Other ways of looking at the "search" issue; "insight" problems

- Retrieving schemas/searching for patterns
- Restructuring the problem space itself
- Functional fixedness and the "Einstellung" effect
Mr. Smith and his wife invited four other couples for a party. When everyone arrived, some of the people in the room shook hands with some of the others. Of course, nobody shook hands with their spouse and nobody shook hands with the same person twice.

After that, Mr. Smith asked everyone how many times they shook someone’s hand. He received different answers from everybody.

How many times did Mrs. Smith shake someone’s hand?

From Michalewicz and Fogel, *How to Solve It: Modern Heuristics*
Game-Playing: the Essential Arsenal

• The idea of a “minimax search”
• Avoiding useless search: the alpha-beta algorithm
• Other ways of pruning the search tree
• Other central issues in game-playing
The Game of Sim
In Scheme:

(define (minimax gamestate myfunction otherfunction level)
  (cond ((= level 0) (getvalue gamestate))
        (else
         (let* ((successors (getnextmoves gamestate))
                (othervals
                 (map (lambda (successor)
                       (minimax successor otherfunction
                                    myfunction (- level 1)))
                     successors))
             (apply myfunction othervals))))
(define (alpha-beta alpha beta level node max?)
  (cond ((= level 0) (static-value node))
    (max? (let ((children (find-next-moves node #t)))
           (max-look-at-each-child-node children alpha beta level)))
    (else (let ((children (find-next-moves node #f)))
            (min-look-at-each-child-node children alpha beta level))))

(define (max-look-at-each-child-node children alpha beta level)
  (cond ((null? children) alpha)
        ((>= alpha beta) alpha)
        (else (let ((next-value
                      (alpha-beta alpha beta (- level 1) (first children) #f))
                    (max-look-at-each-child-node
                     (rest children)
                     (max next-value alpha)
                     beta level))))))

(define (min-look-at-each-child-node children alpha beta level)
  (cond ((null? children) beta)
        ((>= alpha beta) beta)
        (else (let ((next-value
                     (alpha-beta alpha beta (- level 1) (first children) #t))
                     (min-look-at-each-child-node
                      (rest children)
                      alpha (min next-value beta)
                      level))))))
Calling alpha-beta with starting values of a (very low) alpha value, a (very high) beta value, a depth of 5, the starting game state, and from the point of view of the maximizer.

(alpha-beta -1000 1000 5 initial-game-state #t)