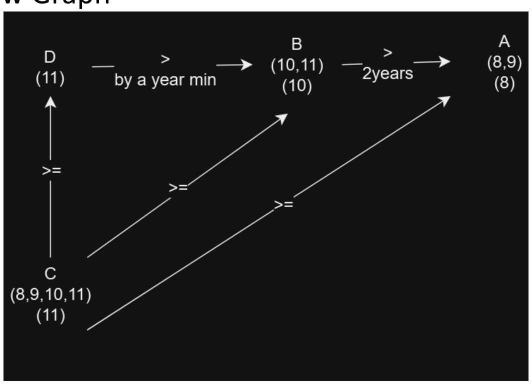
Daniel Willard
CS 471 AI
Prof. Thanh H. Nguyen

Q1) Solving CSPs
given:
n a combined 3rd and 5th grade class, students can be 8, 9, 10, and 11 years old. We are trying to solve for the ages of Ann, Bob, Claire, and Doug (abbreviations: A, B, C, D). Consider the following constraints:
• No students is older in years than Clair (but may be the same age)
• Bob is two years older than Ann
• Bob is younger in years than Doug

1) Draw Graph
   a) 

2) Suppose we are using the AC-3 algorithm for arc consistency. How many total arcs will be en-queued when the algorithm begins execution?
   a) A-> B, B->D, C->A, C->B, C->D. There will be a total of 5 arc to enqueue when the algorithm begins but since there is no direction assocatied there will be a totally of 10 arc 5(connections) * 2(directions).
3) Assuming all ages {8, 9, 10, 11} are possible for each student before running arc consistency, manually run arc consistency on only arc from A to B.
   a) Need to ensure that Ann's age (A) is at least two years younger than Bob's age (B).
      i) Possible ages for A: {8, 9, 10, 11}
      ii) Possible ages for B: {8,9,(10, 11)} (since Ann's age must be two years younger).
      iii) After applying arc consistency, A can only take the values {8, 9},
      iv) as A = 10 and A = 11 would violate the constraint.
   b) Since A can only be 8 or 9, B can be any age from the set {10, 11}. As shown in the graph above
   c) After this operation, there are no more arcs to be processed because we have already considered the only arc, which is A to B. So, C->A, B->A. update all edges to A

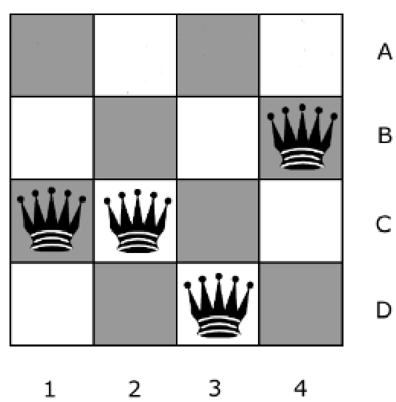
4) Suppose we enforce arc consistency on all arcs. What ages remain in each person's domain?
   a) 8 = Ann

b) 10 = Bob
c) 11 = Claire
d) 11 = Doug

Q2) 4-Queens
Given:

The min-conflicts algorithm attempts to solve CSPs iteratively. It starts by assigning some value to each of the variables, ignoring the constraints when doing so. Then, while at least one constraint is violated, it repeats the following: (1) randomly choose a variable that is currenly violating a constraint, (2) assign to it the value in its domain such that after the assignment the total number of constraints violated is minimized (among all possible selections of values in its domain).

In this question, you are asked to execute the min-conflicts algorithm on a simple problem: the 4-queens problem in the figure shown below. Each queen is dedicated to its own column (i.e. we have variables Q_1, Q_2, Q_3, and Q_4 and the domain for each one of them is $\{A, B, C, D\}$). In the configuration shown below, we have $Q_1 = C$, $Q_2 = C$, $Q_3 = D$, $Q_4 = B$. Two queens are in conflict if they share the same row, diagonal, or column (though in this setting, they can never share the same column).



You will execute min-conflicts for this problem three times, starting with the state shown in the figure above. When selecting a variable to reassign, min-conflicts chooses a conflicted variable at random. For this problem, assume that your random number generator always chooses the leftmost conflicted queen. When moving a queen, move it to the square in its column that leads to the fewest conflicts with other queens. If there are ties, choose the topmost square among them.

1) Starting with the queens in the configuration shown in the above figure, which queen will be moved, and where will it be moved to?

a) Q1 and Q2 are in conflict because they share the same row (row C). We will randomly choose one of them. we choose to move Q1 based off of left most bias.
b) We need to find the position for Q1 that results in the fewest conflicts. The possible moves for Q1 are A, B, and D
    i) If Q1 is moved to D, there are still conflicts with Q3 (both are in the same row).
    ii) If Q1 is moved to B, there are still conflicts with Q4.
    iii) If Q1 is moved to A, there are no conflicts with other queens.
c) So, Q1 will be moved to D.
    i) Q1 = A
    Q2 = C
    Q3 = D
    Q4 = B

2) Continuing off of Part 1, which queen will be moved, and where will it be moved to?
    a) Need to find the leftmost conflicted queen. In this case, it's Q2 because it's in the same diagonal as Q3.
    b) The possible moves for Q2 are A, B, and D (to minimize conflicts with other queens).
        i) If Q2 is moved to A, there is a conflict with Q1
        ii) If Q2 is moved to B, there is a conflict with Q1, Q4 (diagonal, row).
        iii) If Q2 is moved to D, there is a conflict with Q3, Q4 (diagonal, row)
    c) So, Q2 will be moved to A.
        i) Q1 = A
        Q2 = A
        Q3 = D
        Q4 = B
3) Continuing off of Part 2, which queen will be moved, and where will it be moved to?
    a) Need to find the leftmost conflicted queen. In this case, it's Q1 because it's in the same row as Q2.
    b) The possible moves for Q1 are B, C, and D (to minimize conflicts with other queens).
        i) If Q2 is moved to B, there is a conflict with Q4 (row)
        ii) If Q2 is moved to C, there is no conflict
        iii) If Q2 is moved to D, there is a conflict with Q3
    c) So, Q1 will be moved to A.
        i) Q1 = C
        Q2 = A
        Q3 = D
        Q4 = B

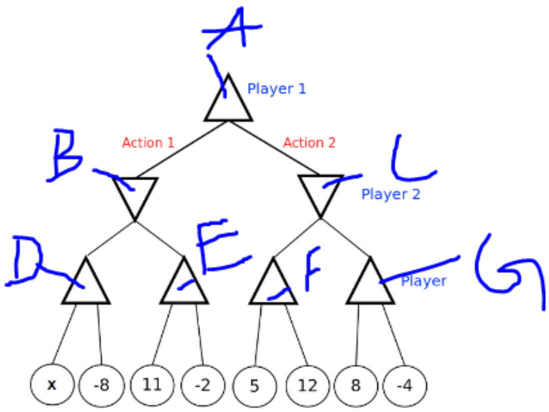
Solution found

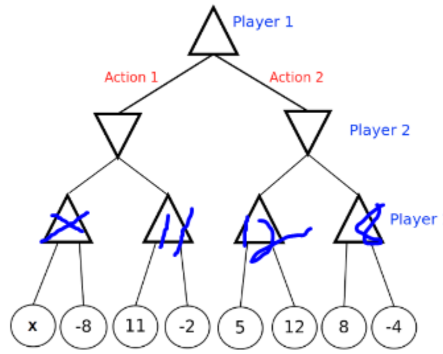
Q3) Minimax and Expectimax

1. Minimax
   - Starting at the leaf nodes, we can see the outcome values for the maximizing player (Player D):
     1. Leaf A: 6
     2. Leaf B: 9
     3. Leaf C: 8
   - Moving up the tree, the minimizing player (Players A, B, C) will choose the lowest value from the child nodes:
     1. Node A: Min(6, 1, 7) = 1
     2. Node B: Min(9, 5, 2) = 2
     3. Node C: Min(3, 8, 4) = 3
   - Continuing up the tree, the maximizing player (Player D) will choose the highest value from the child nodes:
     1. Node D: Max(1, 2, 3) = 3
   - So, the values for the letter nodes are as follows:
     1. Node D: 3
     2. Node A: 1
     3. Node B: 2
     4. Node C: 3
2. Expectimax
   - Start at the leaf nodes (the square nodes). These represent the values at the end of the game:
     1. Leaf A: {33, 3, 9}
     2. Leaf B: {30, 15, 12}
     3. Leaf C: {6, 18, 0}
     4. Leaf D: {21, 27, 24}
   - Move up to nodes A, B, C and D (maximizing player decisions). The maximizing player (Player G) will choose the maximum value among its child nodes. However, at these nodes, we have chance nodes.
     1. Node A (maximizing player decision): Expected value at A = (1/3) * 33 + (1/3) * 3 + (1/3) * 9 Expected value at A = (1/3) * 33 + (1/3) * 3 + (1/3) * 9 = (11) + (1) + (3) = 15
     2. Node B (maximizing player decision): Expected value at B = (1/3) * 30 + (1/3) * 15 + (1/3) * 12 Expected value at B = (1/3) * 30+ (1/3) * 15 + (1/3) * 12 = (10) + (5) + (4) = 19
     3. Node C (maximizing player decision): Expected value at C = (1/3) * 6 + (1/3) * 18 + (1/3) * 12 Expected value at C = (1/3) * 6 + (1/3) * 18 + (1/3) * 0 = (2) + (6) + (0) = 8
     4. Node D (maximizing player decision): Expected value at D = (1/3) * 21 + (1/3) * 27 + (1/3) * 24 Expected value at D = (1/3) * 21 + (1/3) * 27 + (1/3) * 24 = (7) + (9) + (8) = 24
   - Move up to nodes E and F (minimizing player decisions). At these nodes, the minimizing player (Player B) will choose the minimum value among its child nodes:
     1. Node E: Min(15, 19) = 15
     2. Node F: Min(8, 24) = 8

- Finally, move up to the root node G (maximizing player decision). The maximizing player (Player G) will choose the maximum value among its child nodes:
    1. Node G: Max(15, 8) = 8
- So, the values for the letter nodes are as follows:
    1. Node G: 8
    2. Node E: 15
    3. Node F: 8
    4. Node A: 15
    5. Node B: 19
    6. Node C: 8
    7. Node D: 24

Q4. Unknown Leaf Value





1. Player 1 is looking to maximize their outcome, and Player 2 is minimizing it. Therefore, Player 1 will choose Action 1 (move to node B) if the outcome is better than the alternative, which is Action 2 (move to node C).
   - Player 1 will choose Action 1 if and only if the value of node D (which has an outcome of x) is greater than the value of node G less then node F. So, we can write the constraint as:
   - $x > 8$
2. In this case, Player 1 is looking to maximize their expected outcome, considering the random actions of Player 2. Player 1 will choose Action 1 if and only if the expected value of node D is greater than the expected value of node G (8).
   - We can calculate the expected values for both nodes D and G:
   - $x > 9$, though optimally $x > 11$
3. Player 2 (at nodes B and C) will choose the minimum value from the child nodes:
   - Node B: Min(Max(x, -8), Max(11, -2))
   - Node C: Min(Max(5, 12), Max(8, -4))
   - For the minimax value to be worth more than the expectimax value, we need to ensure that the best case for Player 1 under the minimax scenario is greater than the expected value under the expectimax scenario. This means:
     1. Max(Max(x, -8), Max(11, -2)) > Expected Value under Expectimax
     2. Max(Max(x,-8),11) > Expected Value under Expectimax
     3. Expected Value under Expectimax = (1/2(x))) +(1/2(11)) = 5.5+x/2
     4. Max(x, 11) > 55 +x\2
        1. If x is greater than 11
           1. $x > 5.5 + x/2$ ➔ $x > 11$
        2. If x is greater than 11
           1. $11 > 5.5 + x/2$ ➔ $11 > x$
   - So player one will only pick action one if: $x > 11$.
   - So with Expectimax where x= 12, Expectimax = 11.5 thus larger always
   - Thus No value of x for minmax to be worth more then Expectimax.
   - NONE
4. NONE do to the properties of minmax taking the absolute worst case and Expectimax having a choice that is better thus a larger value. So none.

Q5 Alpha-Beta Pruning

1. Enter the values of the labeled nodes
    - Start with initial alpha and beta values:
        1. alpha = $-\infty$
        2. beta = ∞
        3. Node D: V = $-\infty$
    - Visit Node A (left-most unvisited child):
        1. Node A: V = 4
        2. Update beta = min(beta, 4) = 4
    - Visit Node B (left-most unvisited child):
        1. Node B: V = 3
        2. Update beta = min(4, 3) = 3
    - Visit Node C (left-most unvisited child):
        1. Node C: V = 2
        2. Update beta = min(3, 2) = 2
        3. Prune leaf 1
    - Visit Node D (root):
        1. Node D :V =4
        2. Update alpha min(alpha, 4) = 4
    - The values of the labeled nodes are as follows:
        1. Node A: V = 4
        2. Node B: V = 3
        3. Node C: V = 2
        4. Node D (Root): V = 4
2. Leaves node that are punned are as fallows:
    - 5 by V > beta ➔ 5 >3
    - 6 by V > beta ➔ 6 >3
    - 7 by V > beta ➔ 4 >2