CIS 571 - Assignment 2

Ali Hassani

October 24, 2021

Question 1

1.1 Draw the constraint graph.

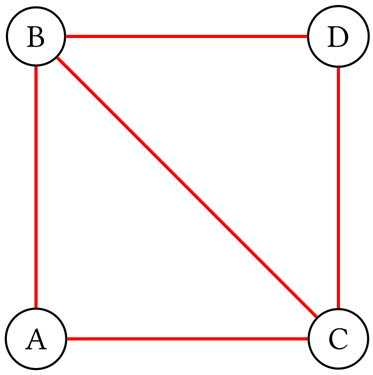


Figure 1: Constraint graph

1.2 Suppose we are using the AC-3 algorithm for arc consistency. How many total arcs will be en-queued when the algorithm begins execution?

5 constraints $\implies$ 10 arcs: `(A, B)`, `(B, A)`, `(A, C)`, `(C, A)`, `(B, C)`, `(C, B)`, `(B, D)`, `(D, B)`, `(D, C)`, `(C, D)`.

1.3 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.

$d_A = \{8, 9, 10, 11\}$
$A = 8 \implies B \in \{10\}$ ✓
$A = 9 \implies B \in \{11\}$ ✓
$A = 10 \implies B \in \{\}$ ✗ $\quad \implies d_A = \{8, 9, 11\}$
$A = 11 \implies B \in \{\}$ ✗ $\quad \implies d_A = \{8, 9\}$

a. Only 8 and 9 remain viable for A.

b. All values technically remain viable, because AC-3 only removes values from the tail domain, and not the head. Therefore, B is not affected.

c. Neighbors of A would have to be rechecked, which are B, C and D. Therefore `(B, A)` `(C, A)` would be added to the queue.

1.4 Suppose we enforce arc consistency on all arcs. What ages remain in each person's domain?

Assuming `(A, B)` is already checked ($d_A = \{8, 9\}$):

`(B, A)` : Values 8 and 9 are invalid for B, therefore they are removed and `(A, B)` , `(C, B)` and `(D, B)` are added to the queue. $d_B = \{10, 11\}$.

`(A, C)` : No violations.

`(C, A)` No violations.

`(B, C)` No violations.

`(C, B)` Values 8 and 9 are invalid for C, therefore they are removed and `(A, C)` , `(B, C)` and `(D, C)` are added to the queue. $d_C = \{10, 11\}$.

`(B, D)` Value 11 is invalid for B, therefore it is removed and `(A, B)` , `(C, B)` and `(D, B)` are added to the queue. $d_B = \{10\}$.

`(D, B)` Values 8, 9 and 10 are invalid for D, therefore they are removed and `(A, D)` , `(B, D)` and `(C, D)` are added to the queue. $d_D = \{11\}$.

`(D, C)` No violations.

`(C, D)` Value 10 is invalid for C, therefore it is removed and `(A, C)` , `(B, C)` and `(D, C)` are added to the queue. $d_C = \{11\}$.

`(B, A)` , `(C, A)` : No violations.

`(A, B)` : Value 9 is invalid for A, therefore it is removed and ... $d_A = \{8\}$.

Skipping the remaining arcs for simplicity.

Remaining: $d_A = \{8\}$, $d_B = \{10\}$, $d_C = \{11\}$, $d_D = \{11\}$.

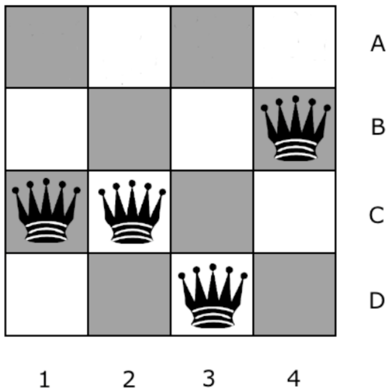
Question 2

1. True. Note: technically backtracking is needed, unless arc consistency is set to return a solution if it finds one (every variable can only take 1 value).
2. True. Note: technically backtracking is needed, unless the checking algorithm is set to return a solution if it finds one (every variable can only take 1 value).
3. False
4. True
5. False
6. False
7. True
8. False
9. True
10. False
11. True
12. False
13. True

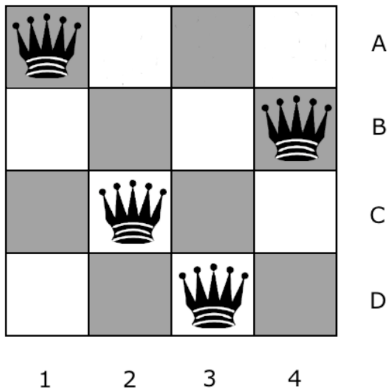
Question 3

3.1 Step 1

The random generator selects the left-most of the violating nodes (queens), which are `Q1`, `Q2`, and `Q3`. `Q4` is not violating any constraints. `Q1` is selected, and the number of violations for `A` through `D` are 0, 1, 1, 2. Therefore it is moved to `A`.



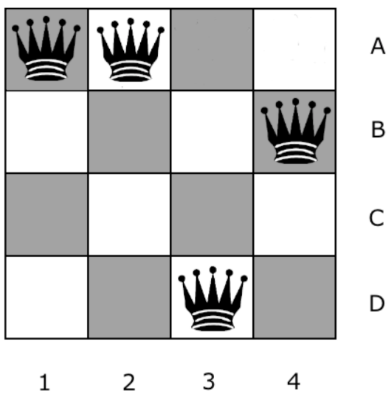
Initial state



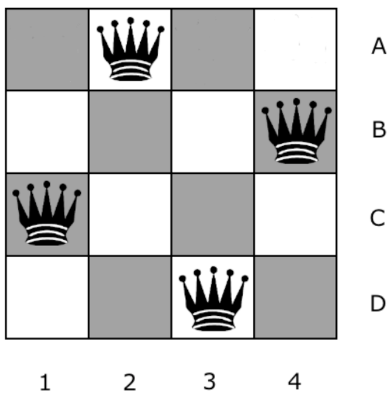
Step 1

3.2 Step 2

The random generator selects the left-most of the violating nodes (queens), which are `Q2`, and `Q3`. `Q2` is selected, and the number of violations for `A` through `D` are 1, 1, 1, 2. Therefore it is moved to `A`.



Step 2



Step 3

3.3 Step 3

The random generator selects the left-most of the violating nodes (queens), which are `Q1`, `Q2`, and `Q3`. `Q1` is selected, and the number of violations for `A` through `D` are 1, 2, 0, 1. Therefore it is moved to `C`.

Question 4

4.1 Search

4.1.1 Minimax Search

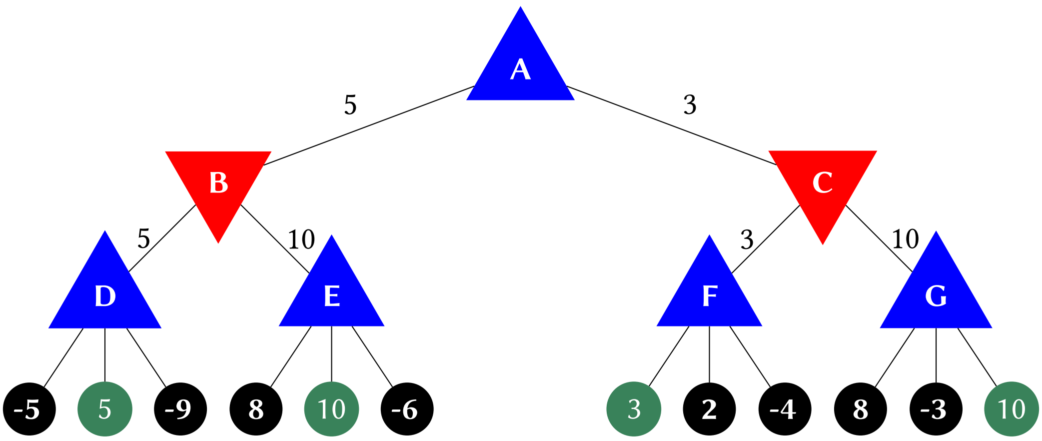


Figure 2: Minimax Search

Root node (`A`) ends up with value 5.

4.1.2 Alpha-Beta pruning

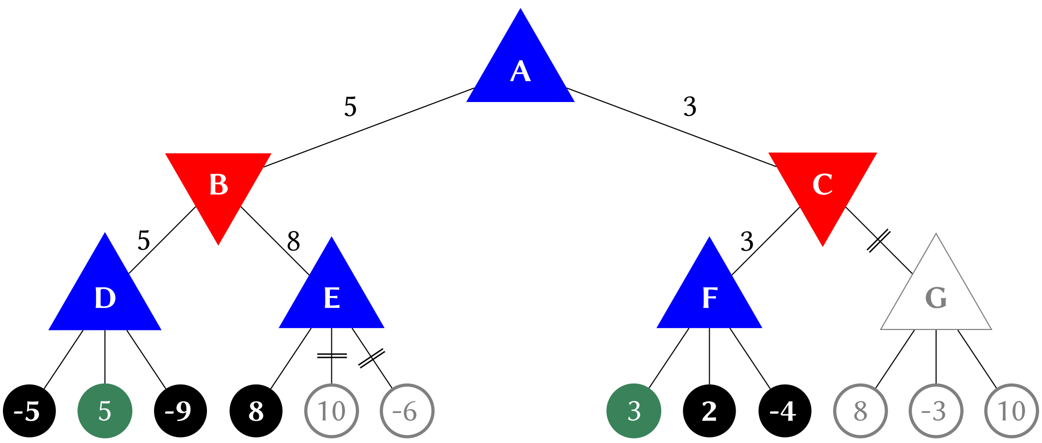


Figure 3: Pruned Tree

4.2 Unknown values

For any $x > 10$ and any $y \leq 10$, given that $x > 10$. . The value at `B` would be $\min(x, 10)$, and given $x > 10$, it would be 10. As a result, any $y \leq 10$ will result in `G` being pruned.

Question 5

a. The root node (min node) is 3, and the three children would be 3, 5, and 7 from left to right.

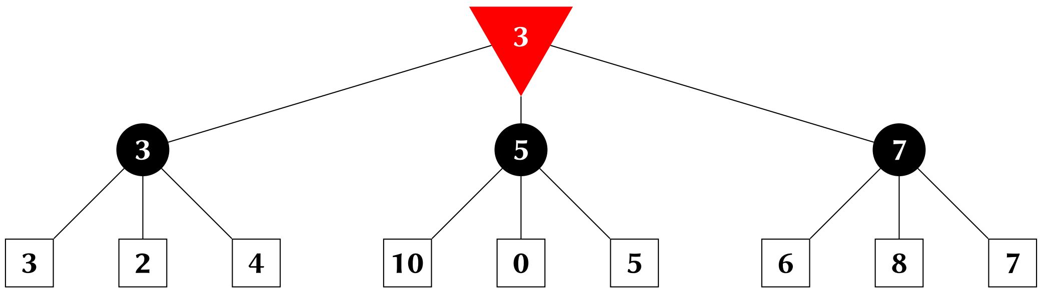


Figure 4: Filled Expectimin Tree

b. Under the assumption that each node has exactly 3 children and all values are nonnegative, leaf nodes 0 and 5 under the second node, and 7 under the right-most node would be pruned. This is because once 10 is explored, even if the other two leaves are 0, the value would end up being ≈ 3.33 which is greater than the 3 we already have explored with the left-most node. Similarly for the rightmost node, $\frac{6}{3} = 2$, so the bounds are still valid, but $\frac{6+8}{3} \approx 4.66$ which is greater than 3, therefore 7 would be pruned.

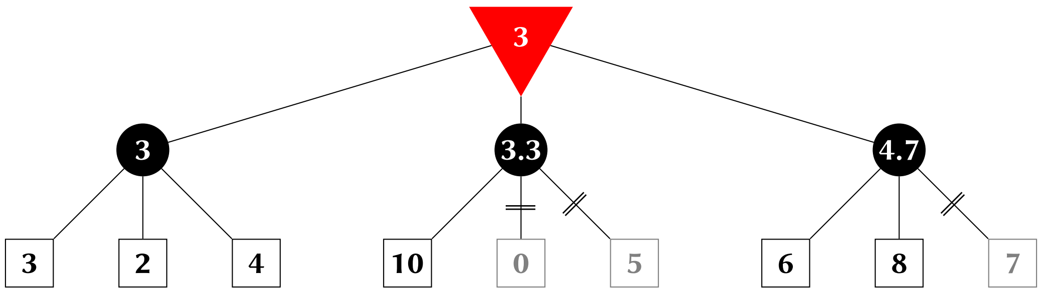


Figure 5: Pruned Expectimin Tree

c. `A` would never be pruned, because it is the last child of the first node that is explored, therefore the bounds are not yet set. For $1 \leq x < 7$, `B` will be pruned. `C` will only be pruned if the leaf to its right is also pruned, and that is given $x < 1$, which means it can only be equal to 0.

Question 6

Pruning cannot take place with respect to the median nodes, because all outcomes should be explored for the median to be calculated.

a. (5) None. V_1 is the first node to be explored, so it cannot be pruned. V_2 cannot be pruned because if it is smaller than V_1, it would be selected by the minimizer. V_3 cannot be pruned because the grandparent node is a median node, V_4 also cannot be pruned because the parent is again a minimizer and requires all children to be explored.

b. (4) V_8. Assuming that $\max(V_5, V_6) < V_7$, then V_8 can be pruned.

c. (4) V_{12}. Assuming that $\max(V_9, V_{10}) < V_{11}$, then V_{12} can be pruned.

d. (5) V_{16}. Assuming that V_{15} is less than the minimum of the two siblings at the minimum level, that means that the smallest of the two siblings is the median, because the output of this section is either V_{15} or V_{16} given that it is less than V_{15}.