

# CS2710 Homework2 Answer Key

---

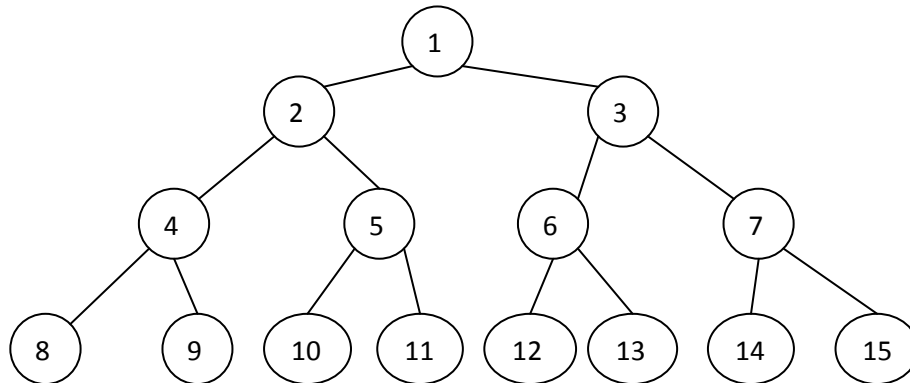
## Problem 1:

A state is a 6 tuple of integers listing the numbers of missionaries, cannibals and boats on the first side, and then the second side of the river. The initial state would be (3,3,1,0,0,0), the goal state would be (0,0,0,3,3,1).

The cost function is one per action, and the successor function of a state is all the states that move 1 or 2 people and 1 boat from one shore to another (as long as the number of cannibals does not exceed the number of missionaries on any shore).

## Problem 2:

A)



The number of nodes at each level would be  $2^n$  where  $n$  is the level, yet the total search space number is infinity since this tree represents all positive integers.

If Depth first was used to explore this search space, it would never return since there are some infinite paths. Breadth first is the way to go when exploring this search space.

B)

Breadth first: 1,2,3,4,5,6,7,8,9,10,11

Depth first: 1,2,4,8,9,5,10,11

Iterative deepening: 1; 1,2,3; 1,2,4,5,3,6,7; 1,2,4,8,9,5,10,11

## Problem 3

### A)

S: A, D

D: S, A, E

E: D, B, F

F: E, B, G

So the nodes in the generation order would be:

S, A, D, E, B, F, G

### B)

Solution path:

S, D, E, F, G

### C)

The cost of this solution is 15 which is not optimal, because [S, A, B, C, G] path has a cost of 11.

## Problem 4

The sequence of the queue represented by C[g, h, f] where C is the city name.

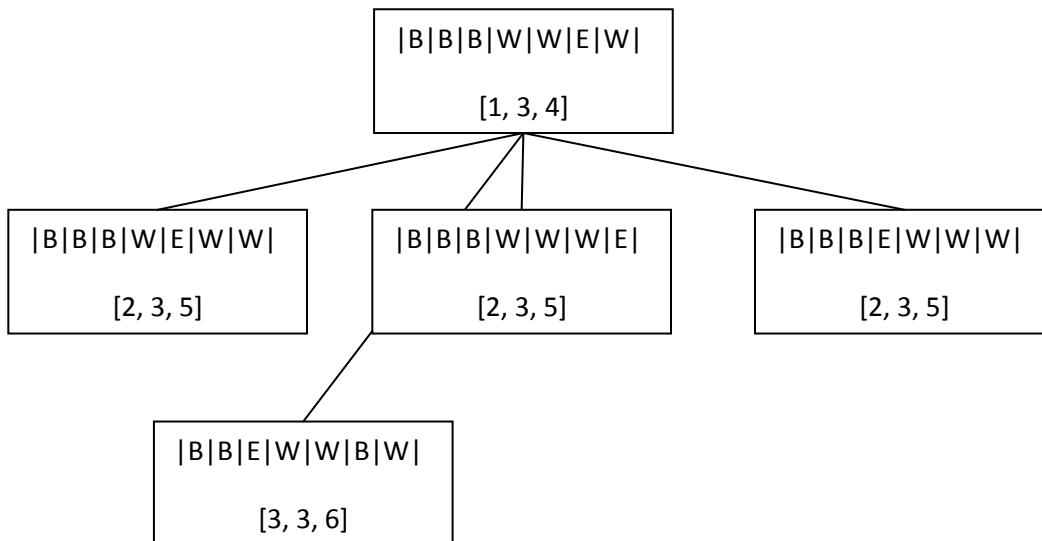
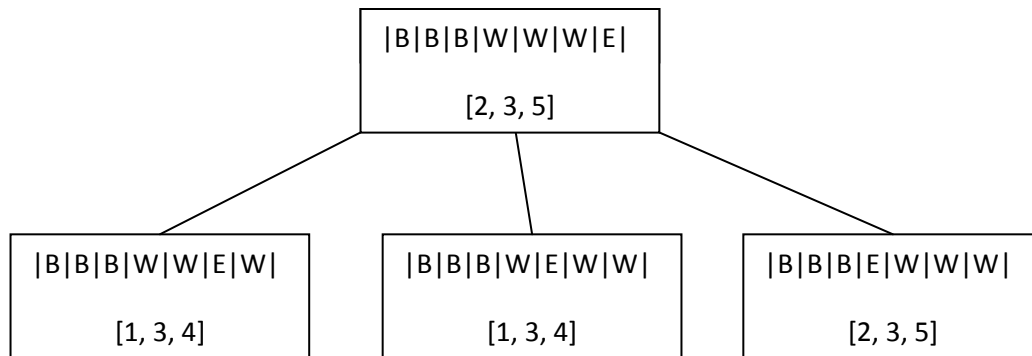
1. L[0,244, 244]
2. M[70, 244, 311], T[111, 329, 440]
3. L[140, 244, 384], D[145, 242, 387], T[111, 329, 440]
4. D[145,242, 387], T[111, 329, 440], M[210, 241, 451], T[251, 329, 580]
5. C[265,160, 425], T[111, 329, 440], M[210, 241, 451], M[220, 241, 461], T[251, 329, 580]
6. T[111, 329, 440], M[210, 241, 451], M[220, 241, 461], P[403, 100, 503], T[251, 329, 580], R[411, 193, 604], D[385, 242, 627]
7. M[210, 241, 451], M[220, 241, 461], L[222, 244, 466], P[403, 100, 503], T[251, 329, 580], R[411, 193, 604], D[385, 242, 627]
8. M[220, 241, 461], L[222, 244, 466], P[403, 100, 503], L[280, 244, 524], D[285, 242, 527], T[251, 329, 580], A[229, 366, 595], R[411, 193, 604], D[385, 242, 627]
9. L[222, 244, 466], P[403, 100, 503], L[280, 244, 524], D[285, 242, 527], L[290, 244, 534], D[295, 242, 537], T[251, 329, 580], A[229, 366, 595], R[411, 193, 604], D[385, 242, 627]
10. P[403, 100, 503], L[280, 244, 524], D[285, 242, 527], M[292, 241, 533], L[290, 244, 534], D[295, 242, 537], T[251, 329, 580], A[229, 366, 595], R[411, 193, 604], D[385, 242, 627], T[333, 329, 662]
11. B[504,0,504], L[280, 244, 524], D[285, 242, 527], M[292, 241, 533], L[290, 244, 534], D[295, 242, 537], T[251, 329, 580], A[229, 366, 595], R[411, 193, 604], D[385, 242, 627], T[333, 329, 662], R[500, 193, 693], C[541, 160, 701]

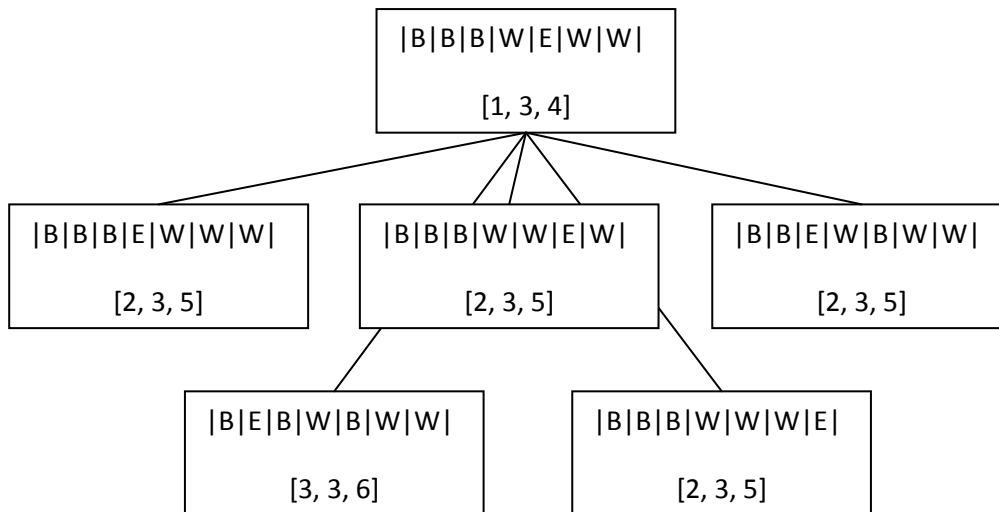
## Problem 5

A)

The given heuristic is admissible because those white tiles would need to move to the left of the leftmost black tile to have a goal state. The cost of doing this at least equals the number of white tiles for the black tile to hop over.

B)





## Problem 6

### A)

H2 would be stuck at local maxima (a node whose value is higher than all its neighbors) after the first move. H1 on the other hand, is not stuck on local maxima so it can reach the global maxima (the solution).

### B)

Let  $\{b, c, d, e, f, g, h, a\}$  represents the problem initial state where all these block are on the same stack where b is the bottom block (the one on the table), and  $\{b, c, d, e, f, g, h, a\}$  represent another state where we have 2 stacks one containing a and the other one containing all the rest of the blocks.

Initial State:  $\{b, c, d, e, f, g, h, a\}$

Move 1:  $\{b, c, d, e, f, g, h, a\}$

Move 2a:  $\{b, c, d, e, f, g, a, h\}$  (put h on top of a)

Move 2b:  $\{b, c, d, e, f, g, a, h\}$  (put h on table)

H1:

H1(goal) = 28

H1(initial state) = -28

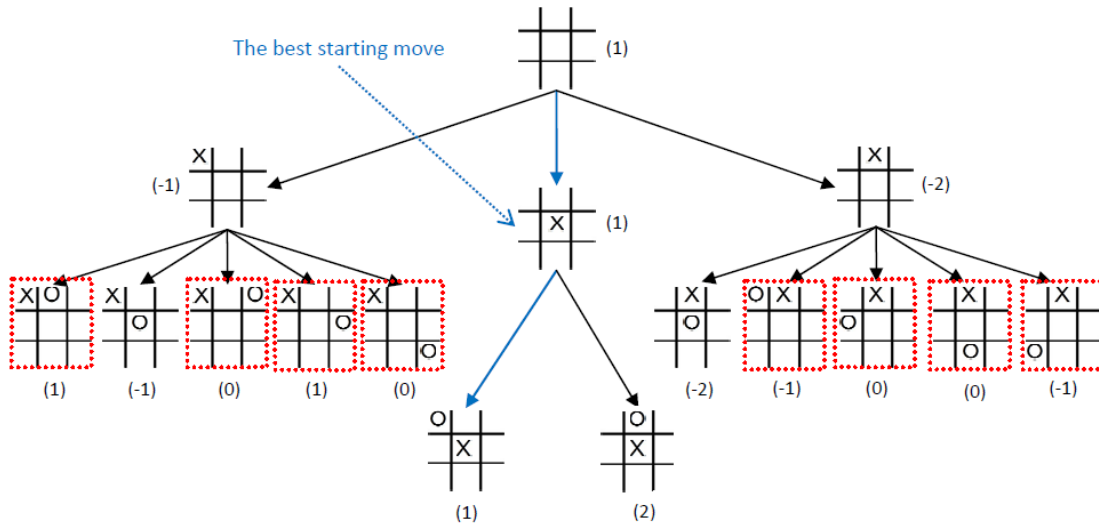
H1(move 1) = -21

H1(move 2a) = -16

H1(move 2b) = -15

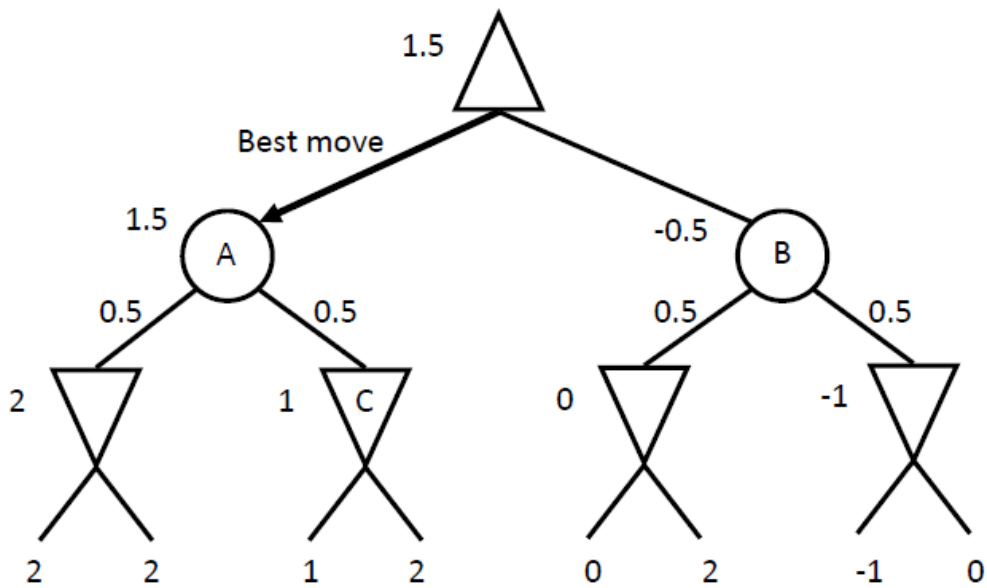
H2:  
H2(goal) = 8  
H2(initial state) = 4  
H2(move 1) = 6  
H2(move 2a) = 4  
H2(move 2b) = 4

**Problem 7** [Figure took from Huy Viet Nguyen solution]



The previous figure shows the game tree, the evaluation value below each board. The leaves with parameter are the ones that do not need to be evaluated, assuming optimal ordering.

**Problem 8** [Figure took from Huy Viet Nguyen solution]



## B)

False: if last 2 leaves were +infinity then the values of the min node and chance node above would also be +infinity, and the best move would change.

True: even if the last node is +infinity the min node would still be -1 and the best move would still be the same.

## C)

If both leaves have a value of -2 then value of  $c = -2$  and  $a = 0$

If both leaves have a value of 2 then value of  $c = 2$  and  $a = 2$

So  $0 \leq a \leq 2$

## Problem 9

### Solution A

- There is a variable corresponding to each of the  $n^2$  positions on the board
- Each variable can take 2 values {occupied, vacant}
- Every pairs of squares separated by a knight's move is constrained, such that both cannot be occupied. Furthermore, the entire set of square is constrained, so that the total number of occupied squares is  $k$ .

### Solution B

- There is a variable corresponding to each knight
- Each variable domain is the set of squares
- Every pair of knights is constrained, such that no two knights can be on the same square or on squares separated by a knight's move.

## Problem 10

### A)

Remaining Domain after enforcing arc consistency

$A = \{2,3\}$ ,  $B = \{2,3\}$ ,  $C = \{1,2\}$ ,  $D = \{2,3\}$

### B)

After  $A=2$

$A = \{2\}$ ,  $B = \{3\}$ ,  $C = \{1\}$ ,  $D = \{2\}$

### C)

$A=2$ ,  $B=3$ ,  $C=1$ ,  $D=2$

### D)

False, even with arc consistency we can backtrack while solving a CSP.