Homework #5

Chapter 4

1. The sliding-tile puzzle consists of three black tiles, three white tiles, and an empty space in the configuration shown below:

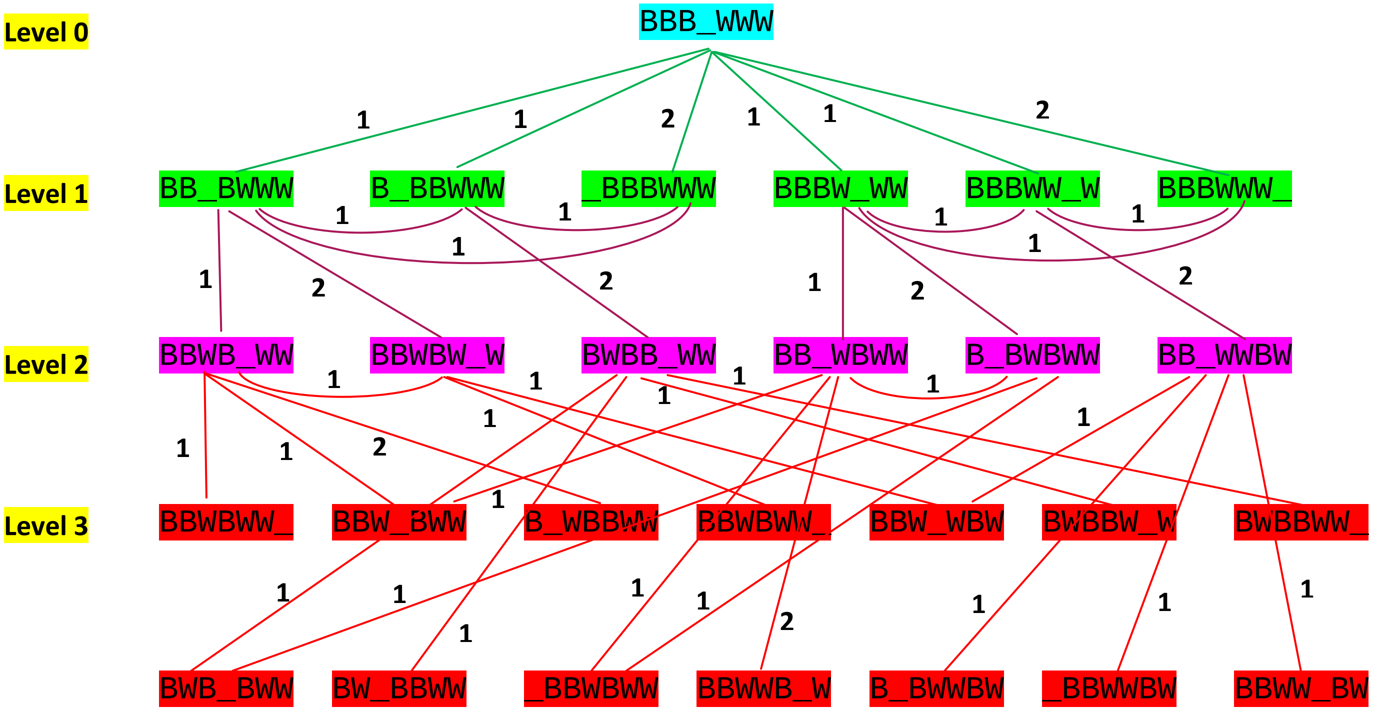
B	B	B		W	W	W

The puzzle has two legal moves with associate costs:

1) A tile may move into an adjacent empty location. This has a cost of 1.
2) A tile may hop over one or two other tiles into the empty position. This has a cost equal to the number of tiles jumped over.

The goal is to move all the white tiles to the left of all the black tiles. The position of the blank is not important.

a) Draw the first three levels of the state space of the sliding-tile puzzle. When you generate the children of a node, make sure you add only those children nodes that have not been included to the state space graph. Label each arc with the cost of the move.


Level 0
BBB_WWW
Level 1
BB_BWWW
B_BBWWW
_BBBWWW
BBBW_WW
BBBWW_W
BBBWWW_
Level 2
BBWB_WW
BBWBW_W
BWBB_WW
BB_WBWW
B_BWBWW
BB_WWBW
Level 3
BBWBWW_
BBW_BWW
B_WBBWW
BBWBWW_
BBW_WBW
BWBBW_W
BWBBWW_
BWB_BWW
BW_BBWW
_BBWBWW
BBWWB_W
B_BWWBW
_BBWWBW
BBWW_BW
1
2

b) Propose a heuristic for solving this problem and analyze it with respect to admissibility, monotonicity and informedness.

h(n) = # of white tiles on the wrong side of each black tile

Breadth-first search is admissible, monotonic and very uninformed.
Counting tiles out of the goal position is a simple useful heuristic that is **admissible**.
A form of this heuristic would be to count all white tiles on the wrong side of each black tile.
For example for the following configuration, there are 3 Ws on the right side of 1st B, and 2 Ws on the right side of both the 2nd and 3rd Bs. Therefore h(n) = 3+2+2 = 7

```
B_WBBWW
```

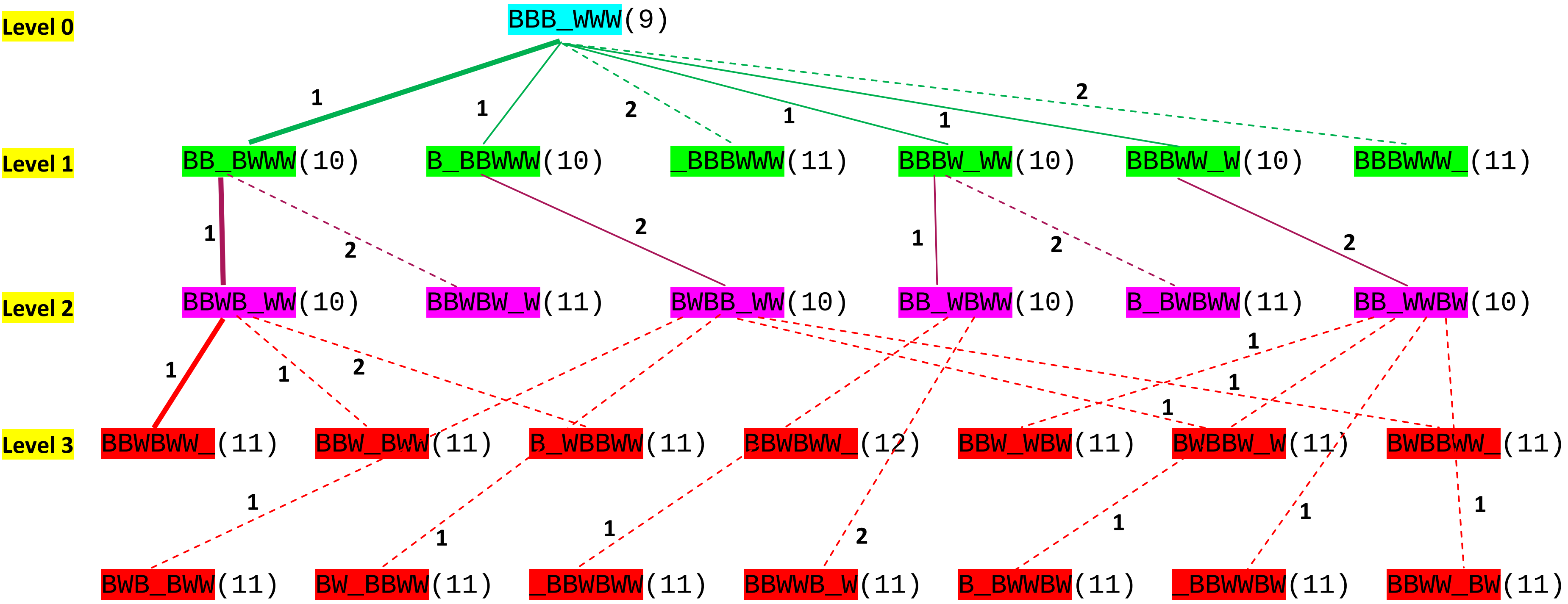
Counting tiles out of place is **not monotonic**, but is **more informed** than breadth-first search. An easy way to prove that a heuristic is non-monotonic is to watch its value change as it goes along a search path.

c) Based on the partial state space you have drawn, find a partial solution path using the best-first search and the evaluation function:

$$f(n) = g(n) + h(n)$$

where g(n) is the cost from the start node to node n
and h(n) is the heuristic estimate of the distance from node n to the goal.

Draw the partial state space that shows your search process (refer to Fig. 4.16 on p-141). You will draw only those nodes that are generated by the best-first search algorithm. If a node is generated but not opened, connect it with a dotted line. If a node is opened, connect it with a solid line. Use the double solid line to show your partial solution path.



Level 0

BBB_WWW(9)

Level 1

BB_BWWW(10) B_BBWWW(10) _BBBWWW(11) BBBW_WW(10) BBBWW_W(10) BBBWWW_(11)

Level 2

BBWB_WW(10) BBWBW_W(11) BWBB_WW(10) BB_WBWW(10) B_BWBWW(11) BB_WWBW(10)

Level 3

BBWBWW_(11) BBW_BWW(11) B_WBBWW(11) BBWBWW_(12) BBW_WBW(11) BWBBW_W(11) BWBBWW_(11)

BWB_BWW(11) BW_BBWW(11) _BBWBWW(11) BBWWB_W(11) B_BWWBW(11) _BBWWBW(11) BBWW_BW(11)