Branch and Bound

15-puzzle problem

1	3	4	15
2		5	12
7	6	11	14
8	9	10	13

(a) An arrangement

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

(b) Goal arrangement

Functions for LCBB

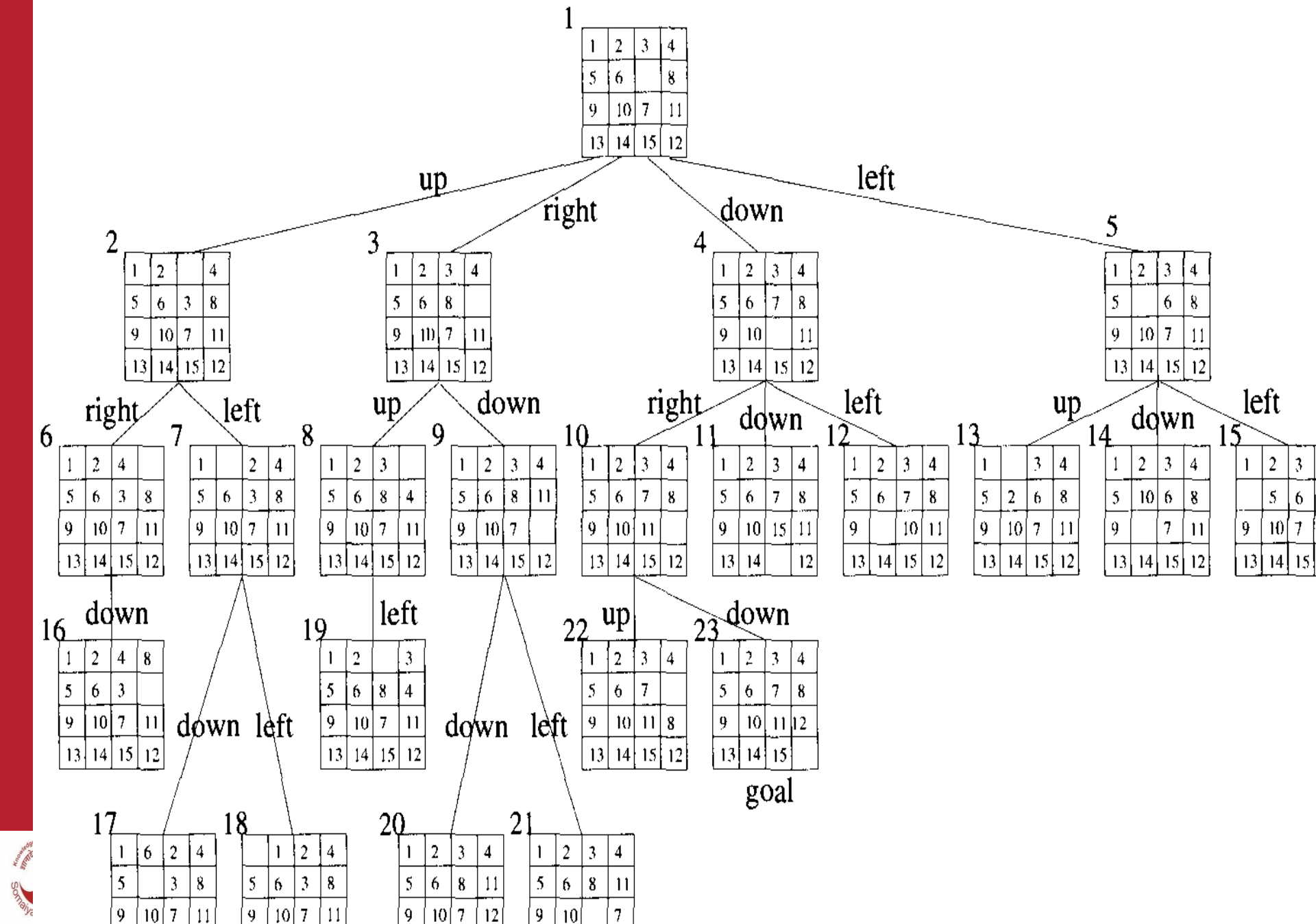
Estimated cost $\hat{c}(x)=f(x)+\hat{g}(x)$

Where

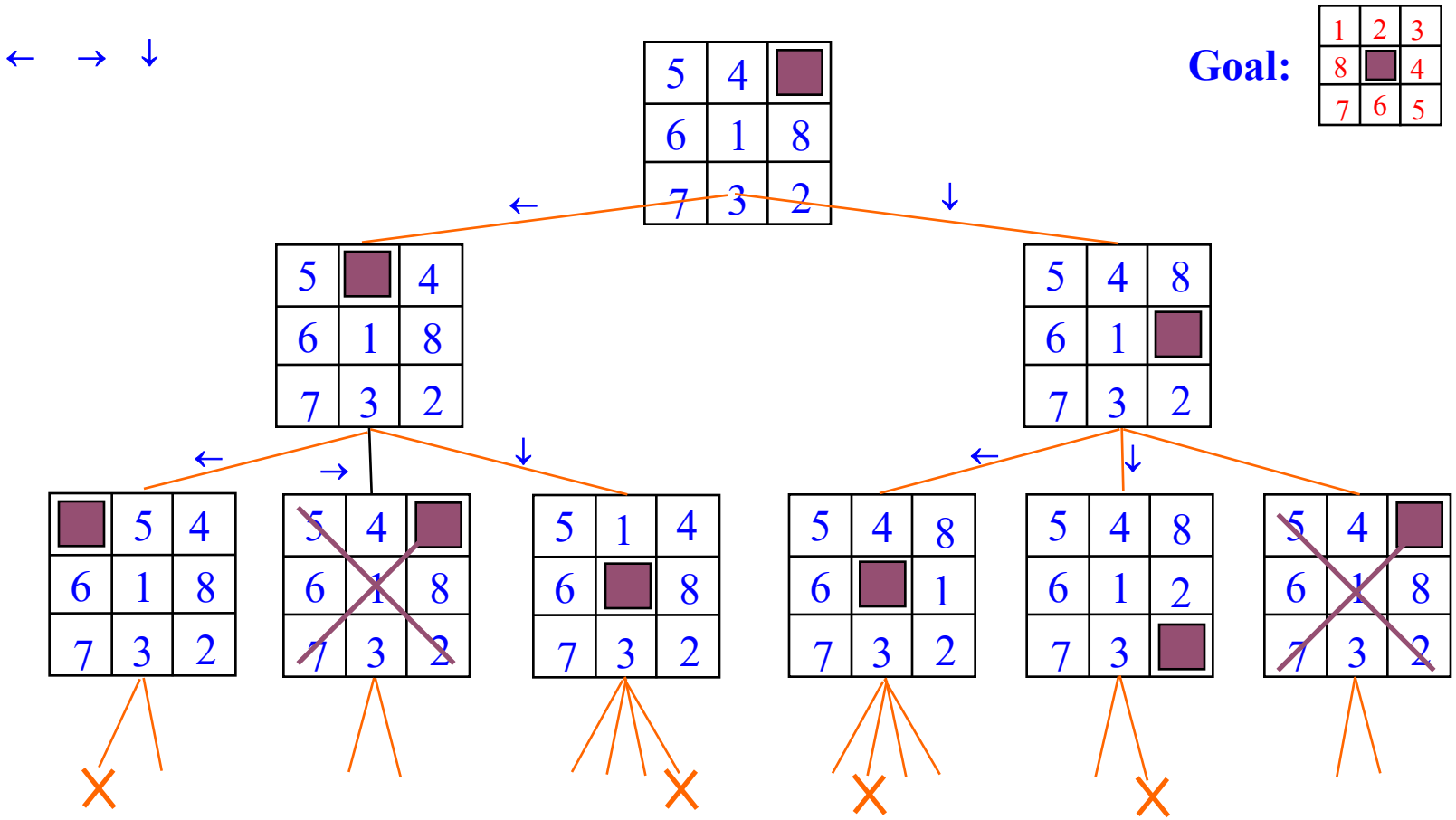
f(x)=length of the path from the root to node x OR no of moves from initial state

$\hat{g}(x)$= Number of non-blank tiles not in the goal position

$\hat{c}(x)$= is the estimated minimum cost to reach the goal node.


1
up
right
down
left
2
3
4
5
right
left
up
down
right
down
left
up
down
left
6
7
8
9
10
11
12
13
14
15
down
left
down
left
up
down
16
17
18
19
20
21
22
23
goal

Solving 8 Puzzle



The 15-puzzle: Cost Estimation

- $\hat{C}(x) = f(x) + \hat{g}(x)$
  - $f(x)$ is the length of the path from the root to node x
  - $g(x)$ is an estimate of the length of a shortest path from x downward to a goal node.
- $\hat{g}(x)$ = number of nonblank tiles not in their goal position

Solving the 15-Puzzle

1	2	3	4
5	6		8
9	10	7	11
13	14	15	12

Initial

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Goal

Solving the 15-Puzzle Contd..

1	2	3	4
5	6		8
9	10	7	11
13	14	15	12

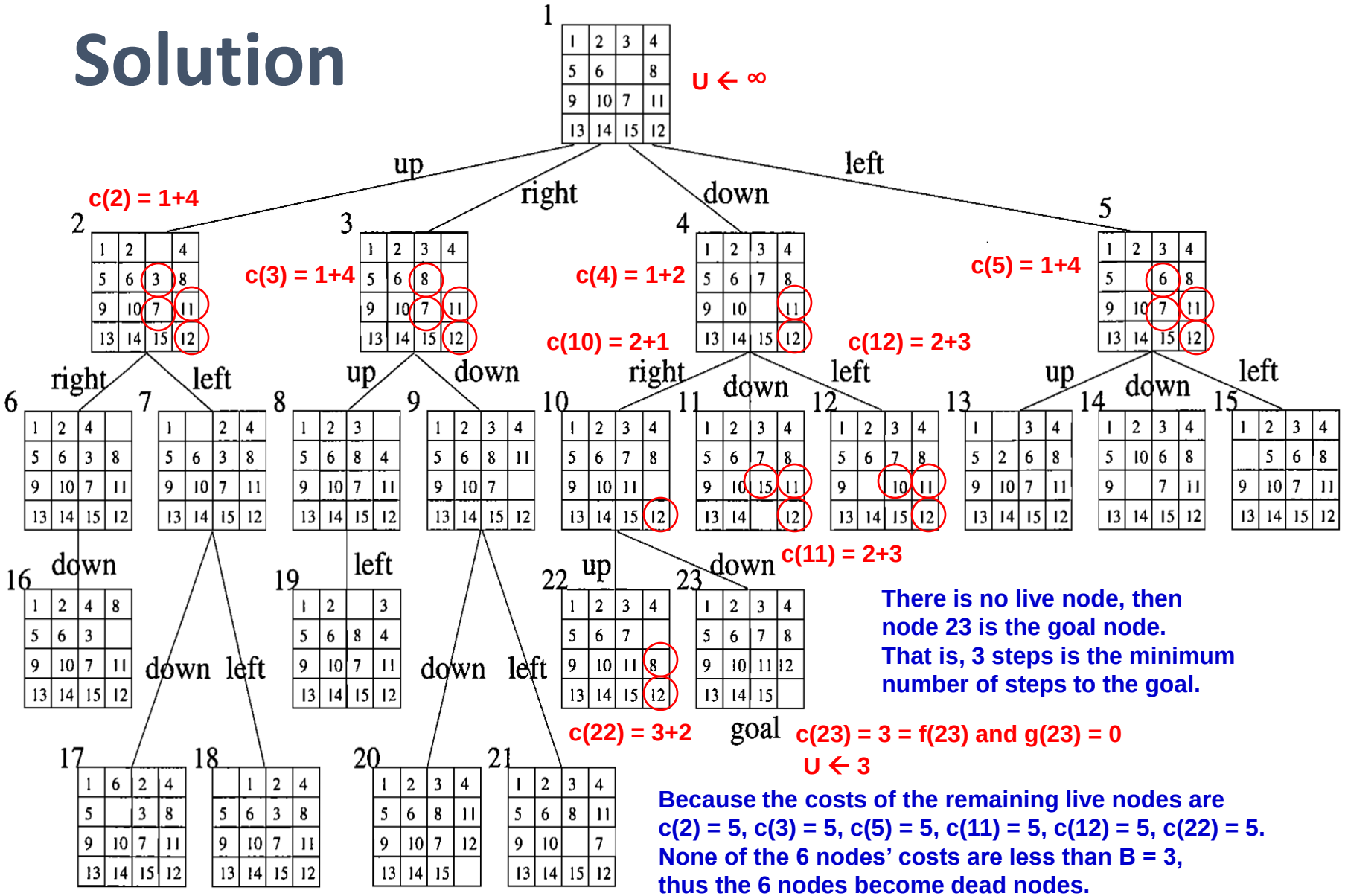
Initial

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Goal

$\hat{g}(x)$= number of misplaced tiles = 3

Solution



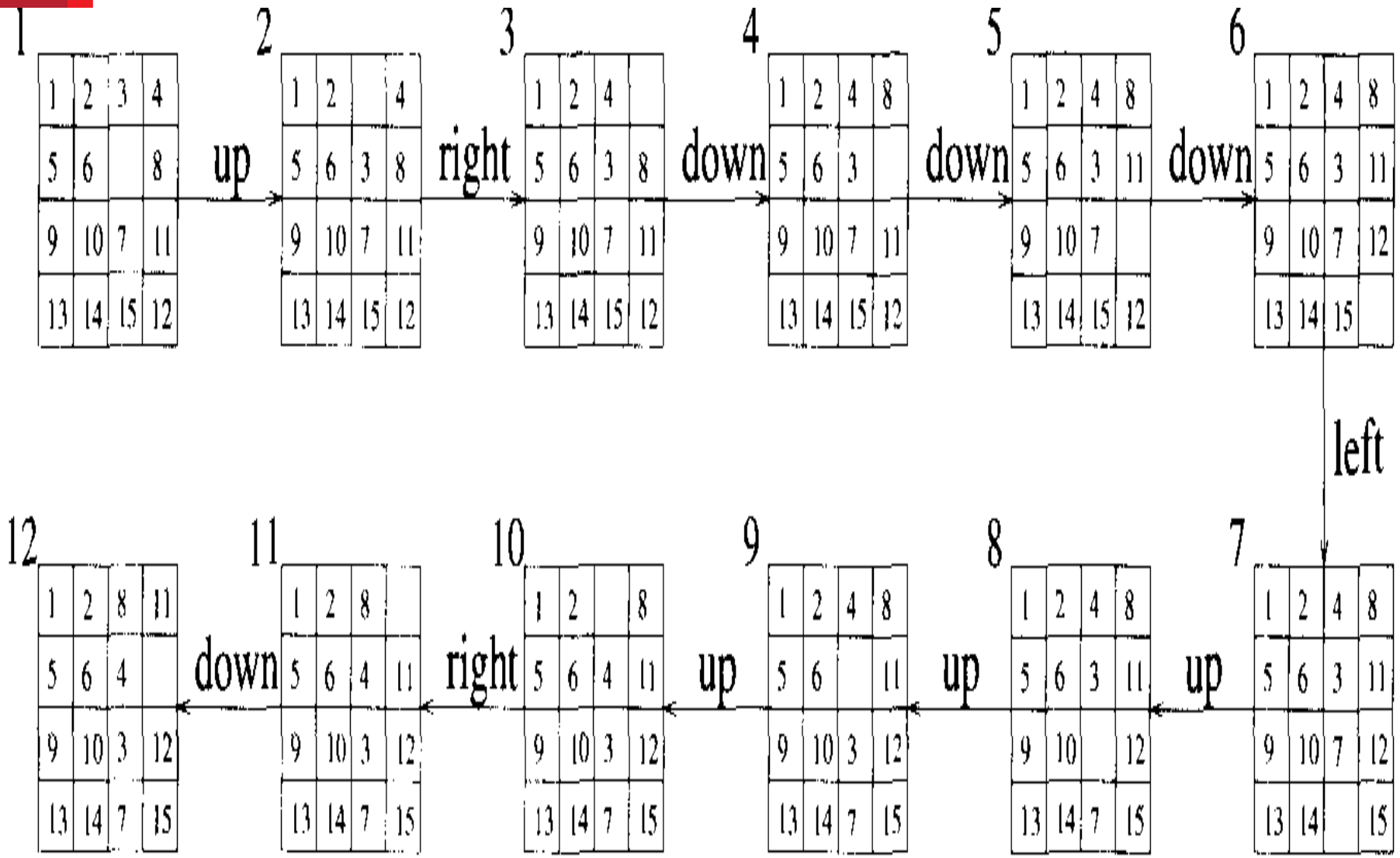
There is no live node, then node 23 is the goal node. That is, 3 steps is the minimum number of steps to the goal.

Because the costs of the remaining live nodes are c(2) = 5, c(3) = 5, c(5) = 5, c(11) = 5, c(12) = 5, c(22) = 5. None of the 6 nodes' costs are less than B = 3, thus the 6 nodes become dead nodes.

Tree traversal

- BFS Vs DFS
- Branch and bound- All state-space searches in which all of the children node of E-node are generated before any other live node becomes E-Node
- Live node:- generated node whose children node aren’t generated
- Dead node:- children nodes processed and probably discarded.

DFS


1
2
3
4
5
6
7
8
9
10
11
12
up
right
down
down
down
left
up
up
up
down
right
up

Functions for LCBB

- Cost function
  - Gives approximation
  - Used as bounding function
- Cost of current node= Cost of reaching this node from root +cost of reaching goal node from current node
- Other types
  - FIFO
  - LIFO

Cost functions for 15 puzzle problem

- Number of correct tiles
- Number of incorrect tiles

```
    listnode = record {
        listnode *next, *parent; float cost;
    }

1   Algorithm LCSearch(t)
2   // Search t for an answer node.
3   {
4       if *t is an answer node then output *t and return;
5       E := t; // E-node.
6       Initialize the list of live nodes to be empty;
7       repeat
8       {
9           for each child x of E do
10          {
11              if x is an answer node then output the path
12                  from x to t and return;
13              Add(x); // x is a new live node.
14              (x → parent) := E; // Pointer for path to root.
15          }
16          if there are no more live nodes then
17          {
18              write ("No answer node"); return;
19          }
20          E := Least();
21      } until (false);
22  }
```