Analysis of Algorithms
CSC 402
2022-23

Subject Incharge

Bidisha Roy

Associate Professor

Room No. 401

email: bidisharoy@sfit.ac.in

BRANCH AND BOUND

The Method

- Branch-and-Bound (BnB) algorithms are based on searches of an associated state space tree for goal states.
- In BnB algorithms, all the children of the E-node (the node currently being expanded) are generated before the next E-node is chosen. (unlike Backtracking where the tree is generated depth wise).
- The traversal is in BFS manner.

The Method

- When the children are generated, they become *live* nodes and are stored in a suitable data structure, say *LiveNodes.*
- Various versions of BnB (Branch and Bound) exist, each differing only in the manner the next E-Node amongst LiveNodes is selected
  - FIFO BnB ☐ LiveNode implemented as queue
  - LIFO BnB ☐ LiveNode implemented as stack
  - LC (Least Cost) BnB ☐ LiveNode implemented as priority queue i.e. the node with the best cost is expanded all others are discarded
- We will be discussing LC BnB only

15 puzzle problem

- 15 numbered tiles on a single square frame of 16 tiles.
- Transform the initial arrangement of tiles into goal arrangement from a series of legal moves.

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

Initial Position

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

Goal Position

15 puzzle problem… contd

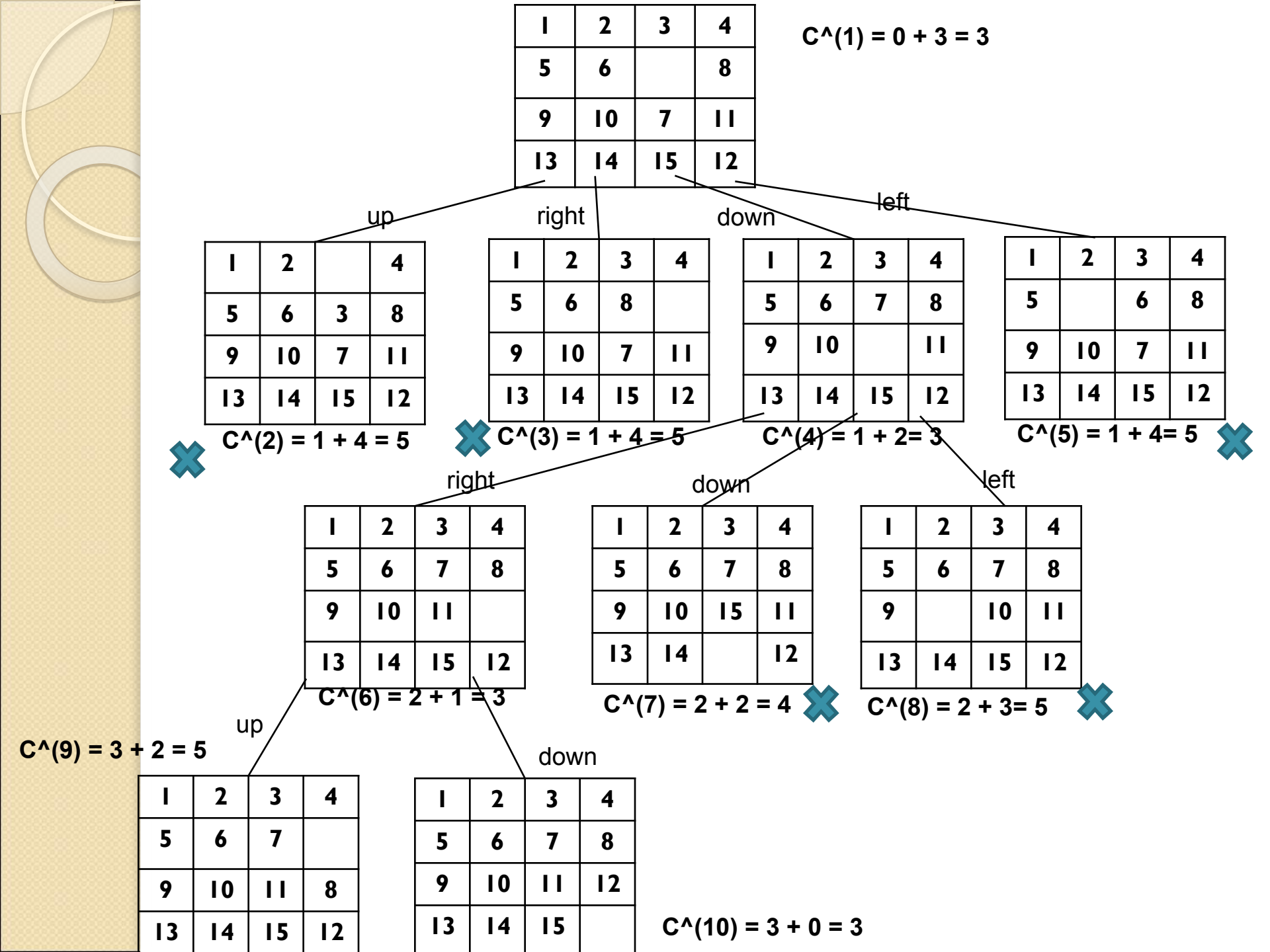
1	2	3	4
5	6		8
9	10	7	11
13	14	15	12

Initial Position

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

Goal Position

- Cost function C^(x) = f(x) + g^(x)
  - C^(x) = estimated cost to reach goal node
  - f(x) = number of moves from initial state
  - g^(x) = estimate of cost of reaching from current state to answer state (number of non blank tiles not in their goal position)


1 2 3 4
5 6 8
9 10 7 11
13 14 15 12
C^(1) = 0 + 3 = 3
up
right
down
left
1 2 4
5 6 3 8
9 10 7 11
13 14 15 12
C^(2) = 1 + 4 = 5
1 2 3 4
5 6 8
9 10 7 11
13 14 15 12
C^(3) = 1 + 4 = 5
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
C^(4) = 1 + 2= 3
1 2 3 4
5 6 8
9 10 7 11
13 14 15 12
C^(5) = 1 + 4= 5
right
down
left
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
C^(6) = 2 + 1 = 3
1 2 3 4
5 6 7 8
9 10 15 11
13 14 12
C^(7) = 2 + 2 = 4
1 2 3 4
5 6 7 8
9 10 11
13 14 15 12
C^(8) = 2 + 3= 5
C^(9) = 3 + 2 = 5
up
down
1 2 3 4
5 6 7
9 10 11 8
13 14 15 12
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15
C^(10) = 3 + 0 = 3

TSP using branch and bound (refer Word document)