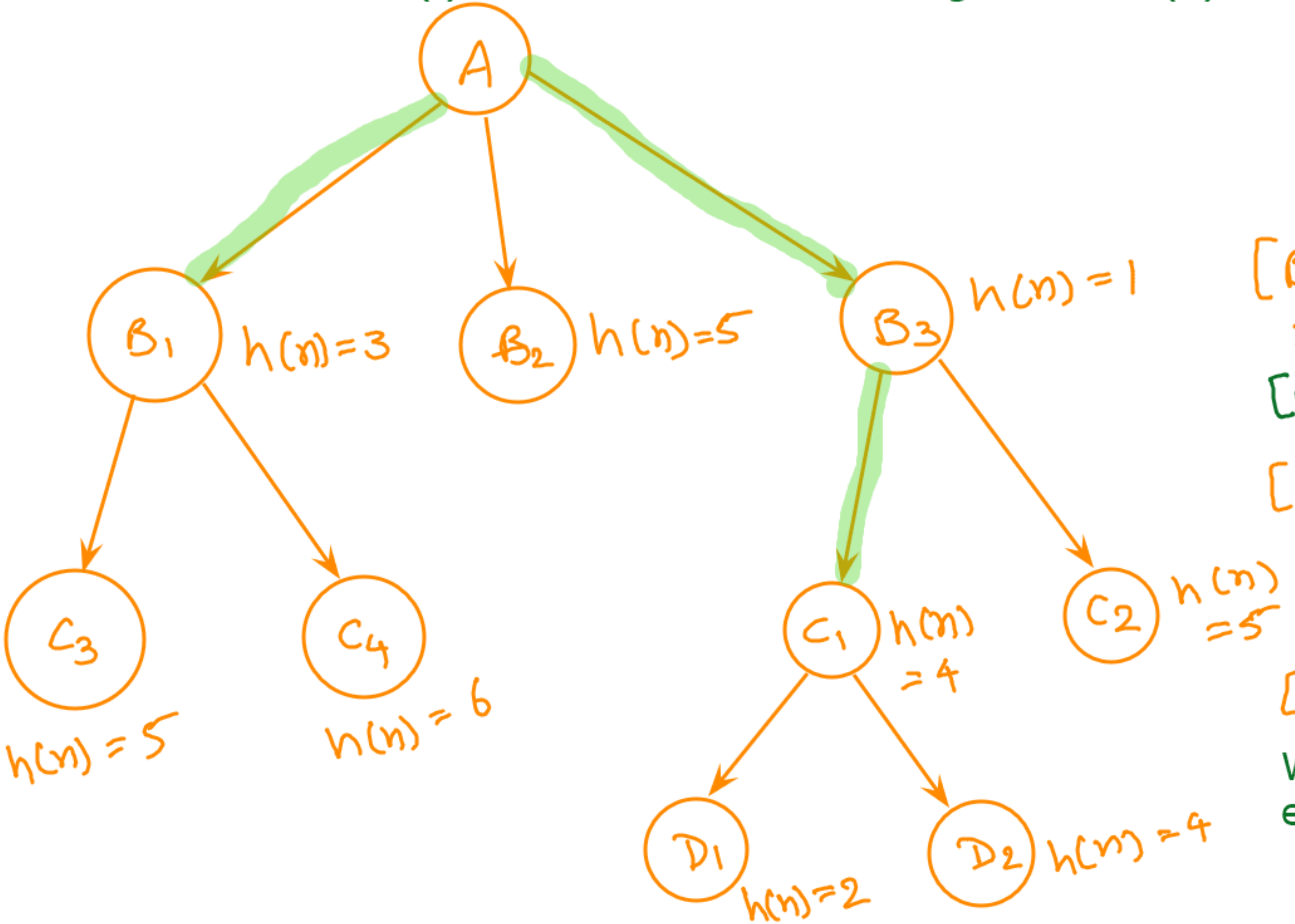
Best First Search Strategies:
(1) BFS & DFS both are exhaustive searches (They expand too many nodes)
(2) Two costs involved- (i) Actual cost to reach the goal node. (ii) Cost for finding the optimal solution.



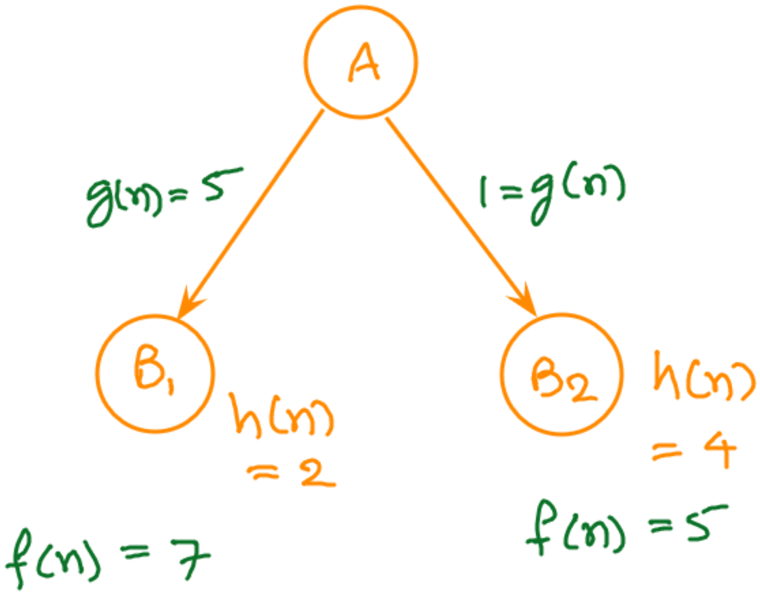
Queue	Open
$[A]$	$[\ ]$
$[B_1, B_2, B_3]$	$[A]$
3 5 1	
$[B_3, B_1, B_2]$	
$[B_1, B_2, C_1, C_2]$	$[A, B_3]$
3 5 4 5	
$[B_1, C_1, B_2, C_2]$	
$[C_1, B_2, C_2, C_3, C_4]$	$[A, B_3, B_1]$

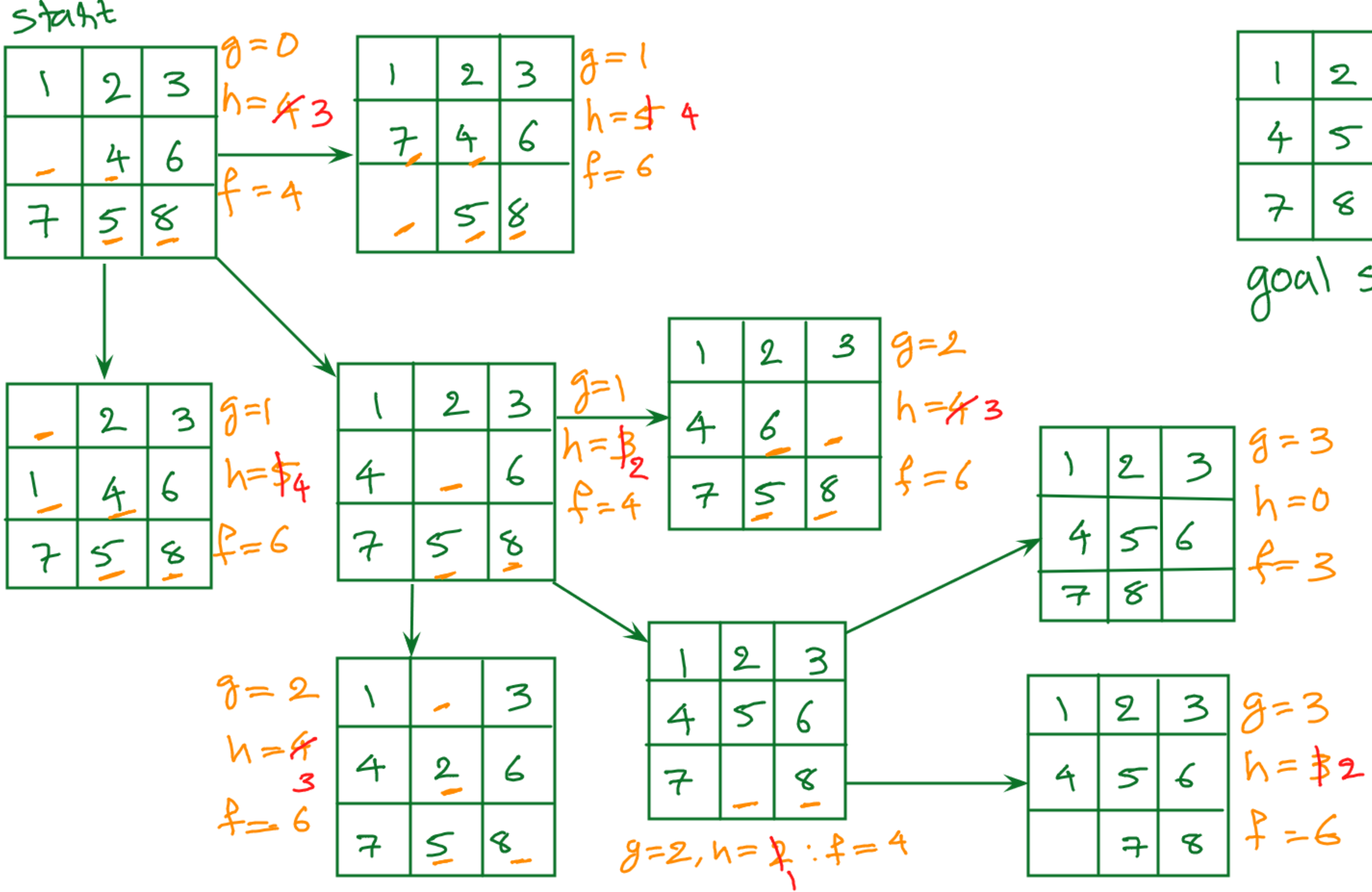
We need to sort the queue every time.

Algorithm 'A': Avoiding paths those are estimated as 'expensive'.

Evaluation Function: g(n) + h(n); where
g(n): cost to reach to state 'n' via optimal path,
h(n): estimated cost to reach from state 'n' to goal state (heuristic value of 'n')

Here, if h(n) is admissible then algorithm is called A*




start
g=0
h=~~4~~ 3
f=4
g=1
h=~~5~~ 4
f=6
goal state
g=1
h=~~5~~ 4
f=6
g=1
h=~~3~~ 2
f=4
g=2
h=~~4~~ 3
f=6
g=3
h=0
f=3
g=2
h=~~4~~ 3
f=6
g=2, h=~~2~~ 1 : f=4
g=3
h=~~3~~ 2
f=6