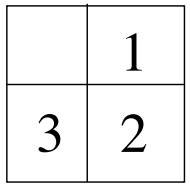
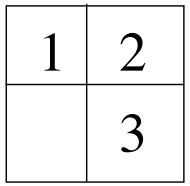
Exercise 1

Consider the following game: three tiles and one blank

Initial State



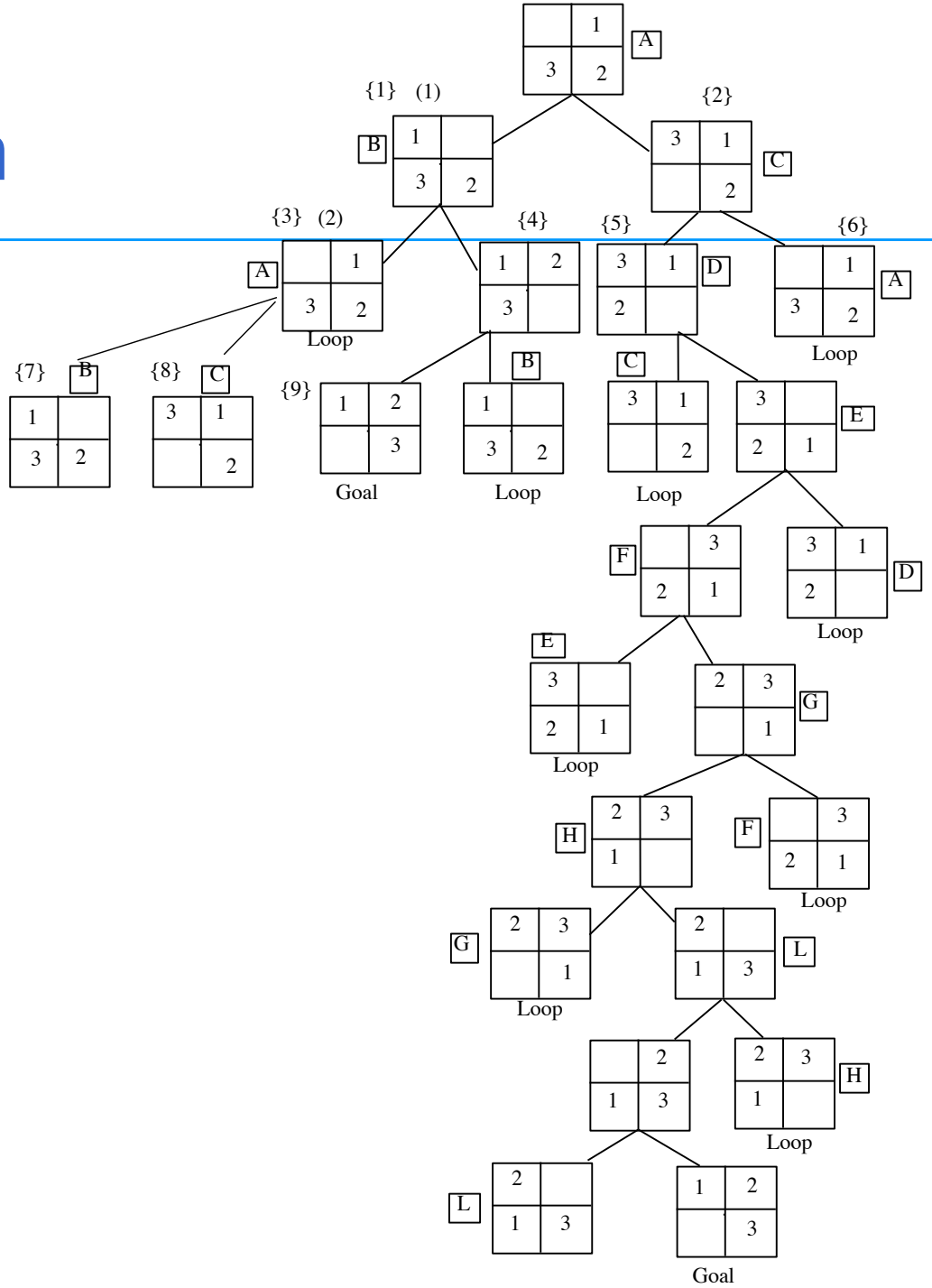


Goal

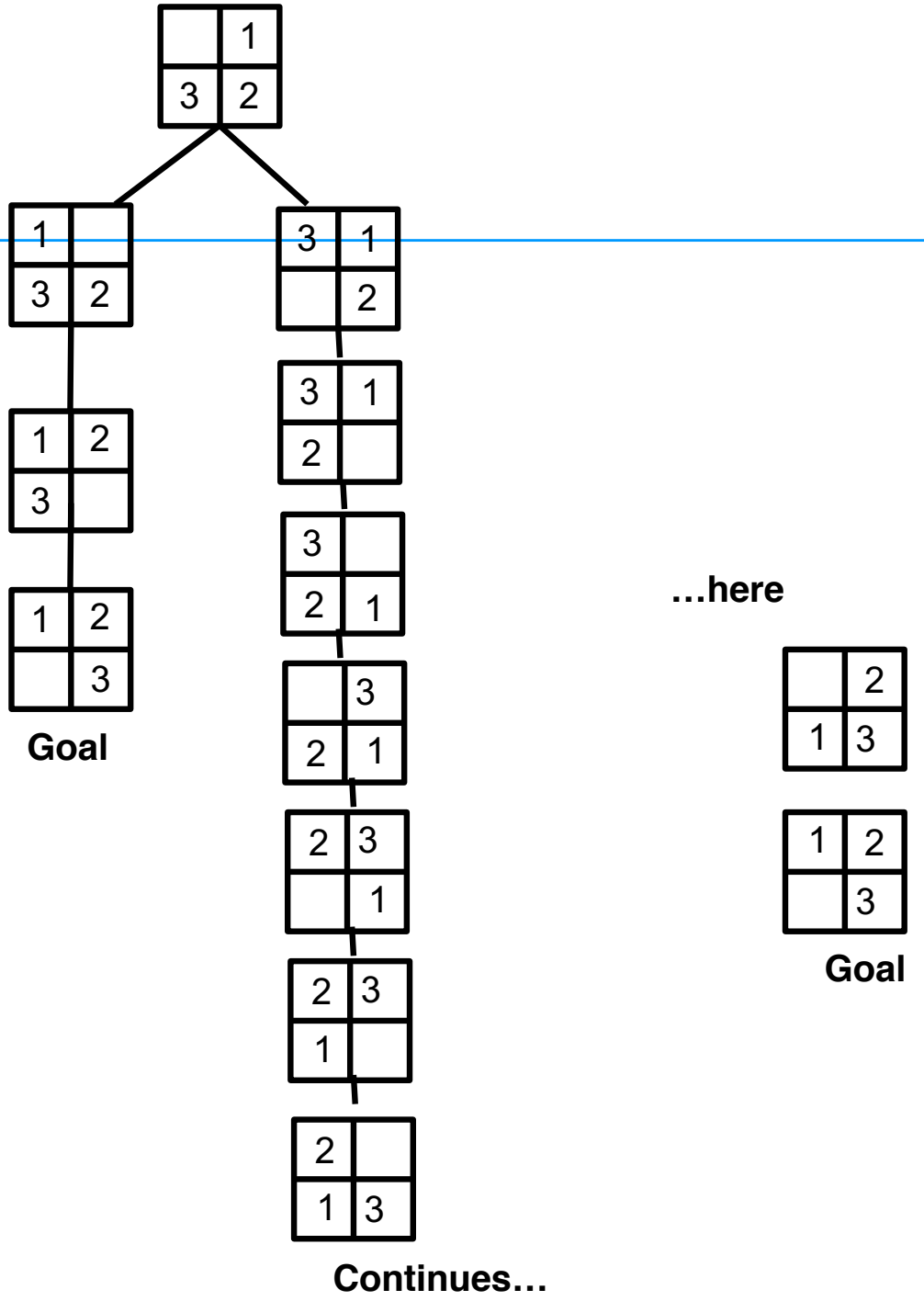
- The operator order is the following: blank moves right, then left, then up and then down. Show the search space generated. After that show how to explore this space with a depth first and then breadth-first.

- Show how the state space changes if we do not allow any move that goes back to a previous state on the same path.

Solution


A
{1} (1)
B
{2}
C
{3} (2)
A
Loop
{4}
{5}
D
{6}
A
Loop
{7}
B
{8}
C
{9}
Goal
B
Loop
C
Loop
E
F
D
Loop
E
G
Loop
H
F
Loop
G
Loop
L
H
Loop
L
Goal

Solution

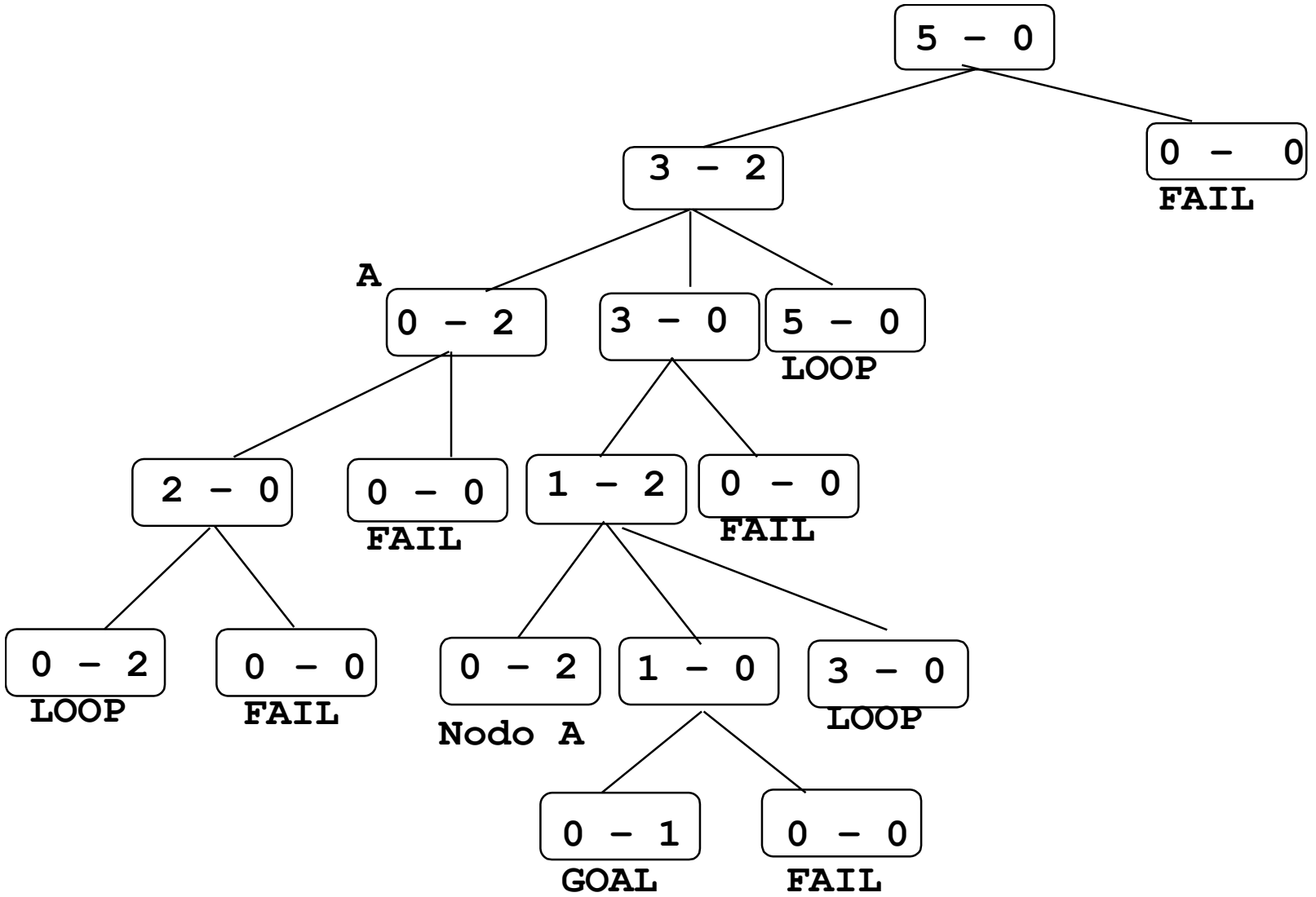


Exercise 2

We have two bins: the first has capacity of 5 liters and is full of water, the second has capacity of 2 liters and is empty.

- We want to obtain one liter of water in the second bin.
- We can transfer water from one bin to the other, we can throw water away, but we cannot obtain new water.
- Show the search space of the problem. Explain after how many steps we reach the goal using breadth-first and depth-first.

Solution

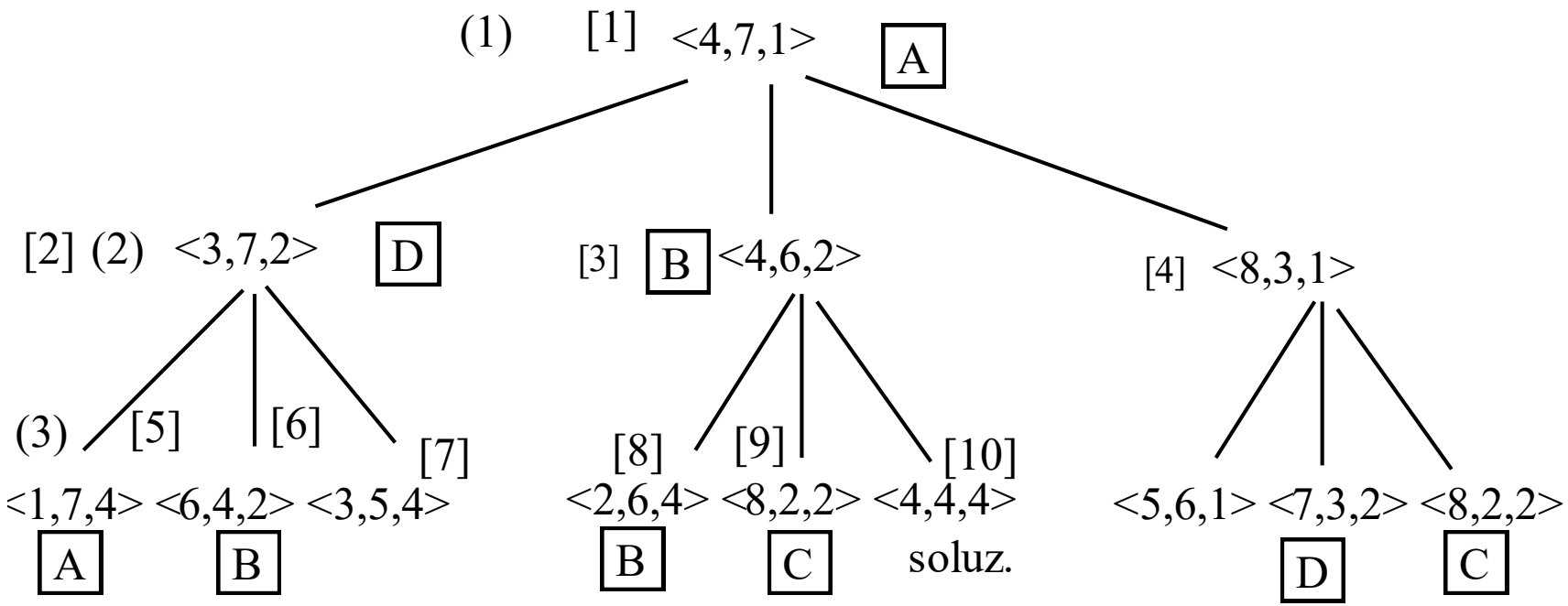

5 – 0
3 – 2
0 – 0
FAIL
A
0 – 2
3 – 0
5 – 0
LOOP
2 – 0
0 – 0
FAIL
1 – 2
0 – 0
FAIL
0 – 2
LOOP
0 – 0
FAIL
0 – 2
Nodo A
1 – 0
3 – 0
LOOP
0 – 1
GOAL
0 – 0
FAIL

Exercise 3

- We have 12 coins in three piles. In the first pile we have 4 coins, in the second we have 7 of them, while the third we have a single coin.
- We have a rule for moving coins: we can move coins from a pile A to a pile B if coins in B double.

- Show the search space starting from the configuration <4,7,1> and the goal is <4,4,4>.

Solution


(1) [1] <4,7,1> A
[2] (2) <3,7,2> D
[3] B <4,6,2>
[4] <8,3,1>
(3) [5] [6] [7]
[8] [9] [10]
<1,7,4> <6,4,2> <3,5,4>
<2,6,4> <8,2,2> <4,4,4>
<5,6,1> <7,3,2> <8,2,2>
A B
B C soluz.
D C

Informed Search Strategies

Remember

- A heuristic function *h(n)* is *admissible if it never overestimates* the cost for reaching the goal

- A heuristic function *h(n)* is *consistent* (*monotone*) if, for each node *n* for every successor *n'* of *n* applying operator *a*, the following holds:

$$h(n) \leq c(n, a, n') + h(n')$$

Exercise 4

- Let us consider the following puzzle:

- We have three black tiles, three white tiles and a blank one.
- Operators are the following
  - A tile can move to a blank one if this is close to it. Cost = 1
  - A tile can move to a blank one jumping over at most two tiles.Cost = distance covered, i.e., the number of jumped tiles plus 1.

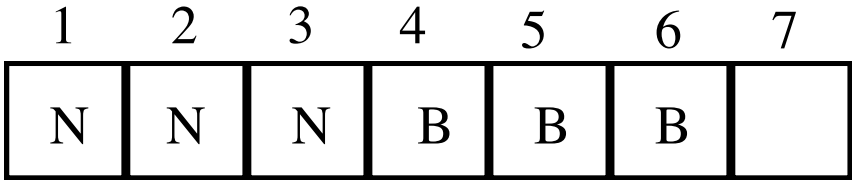
Exercise 4

- The goal is to have all white tiles at the left of black tiles, no matter where the empty tile is.

- Build a heuristic function f(n) = g(n) + h'(n) for each node of this problem and show the search tree generated by the A* algorithm.

Heuristic functions

- Number of tiles out of place: black tiles are out of place if they occupy positions 1, 2, 3 while the white tiles are out of place if they occupy positions 5, 6, 7.



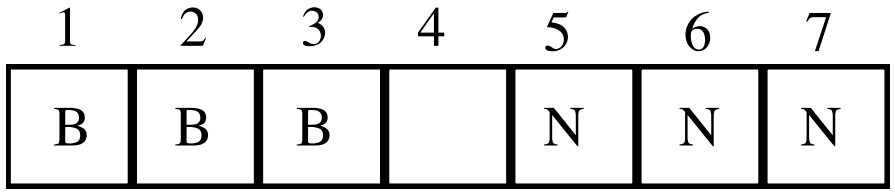
- $h'_1=5$
- Is this admissible?

Heuristic functions

- The distance between black tiles and positions 4,5,6.
- Is this admissible?

Heuristic functions

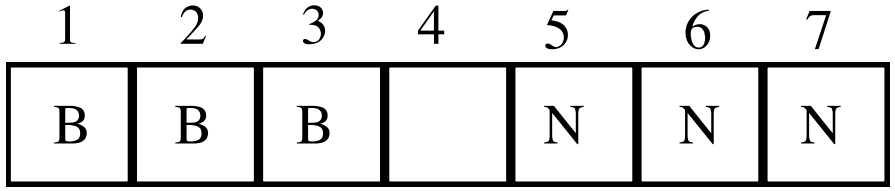
- The distance between black tiles and positions 4,5,6.
- Is this admissible? NO, in the following goal position we have



- h=0 while h'_2=3
- How to correct it?

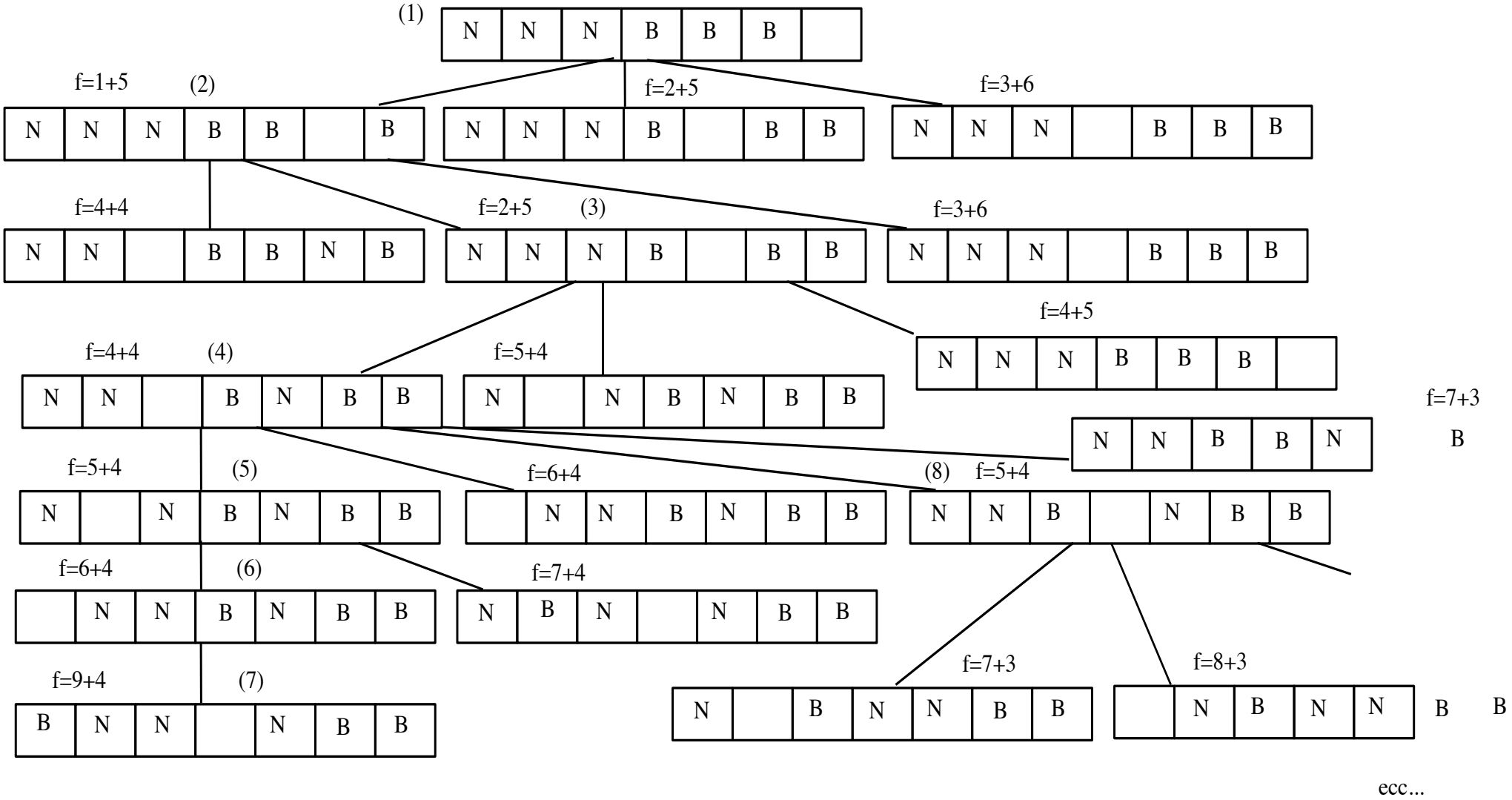
Heuristic functions

- The distance between black tiles and positions 4,5,6.
- Is this admissible? NO, in the following goal position we have

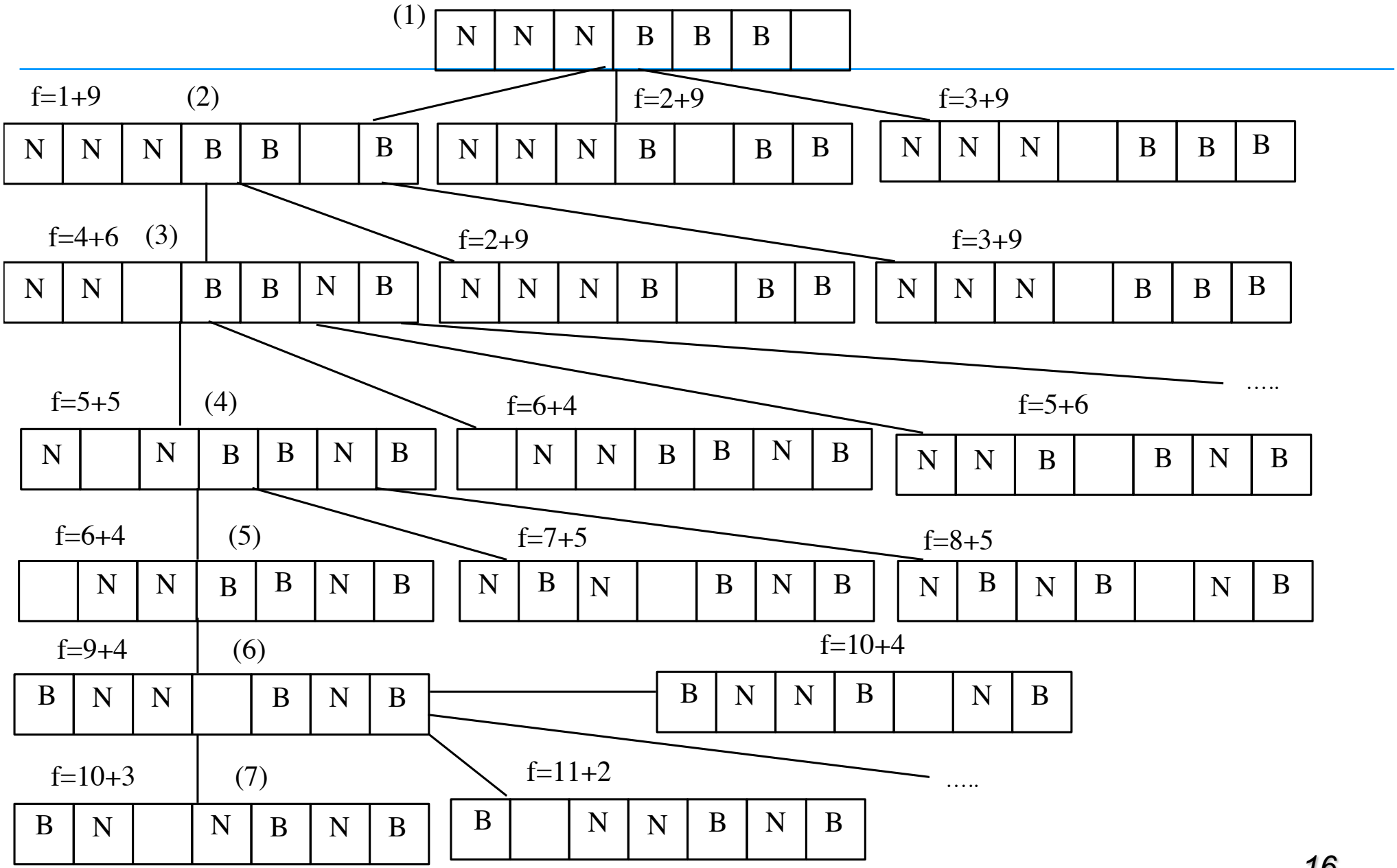


- h=0 while h'_2=3
- How to correct it? We can use the minimum distance between the black tiles and any goal position. We call it h'_3

Search space with heuristic h'_1



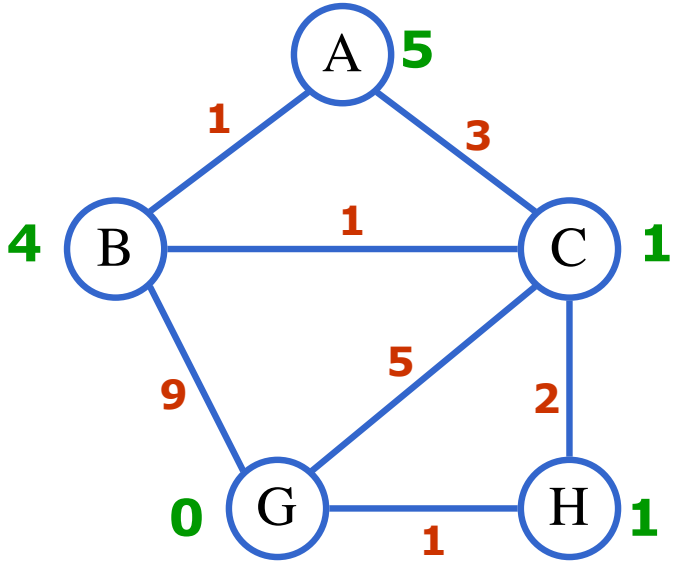
Search space with heuristic h'_3



Exercise 5: search on graphs

- Let us consider the following graph. Red numbers on arcs are costs while green numbers on nodes are heuristic values. Arcs are not oriended so they can be traversed in both directions. Consider A as the initial state and G as a goal.

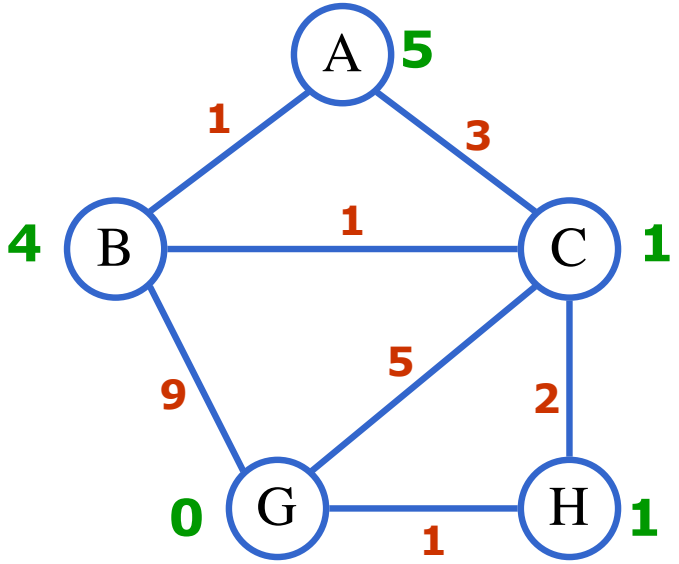
- Show the A* search tree avoiding loops.



Exercise 5: search on graphs

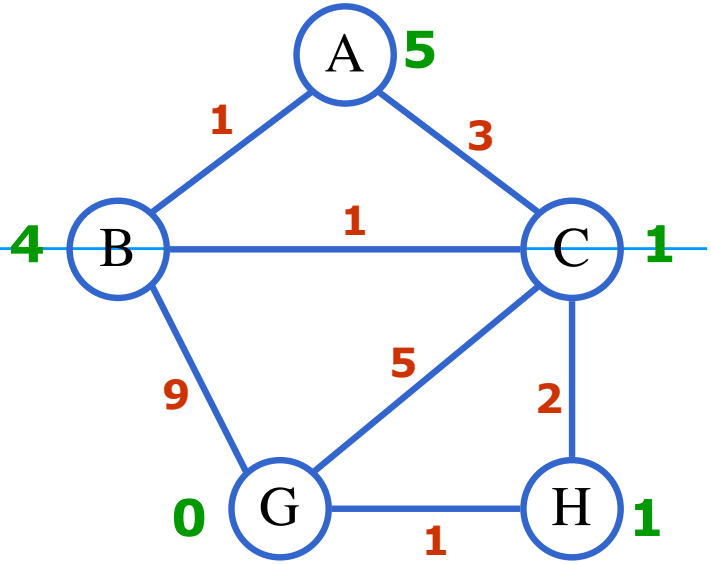
- Let us consider the following graph. Red numbers on arcs are costs while green numbers on nodes are heuristic values. Arcs are not oriended so they can be traversed in both directions. Consider A as the initial state and G as a goal.

- Answer to these questions:
  - Is the heuristic consistent and admissible?
  - Does the algorithm find the minimum cost path to the goal? If not what should I change in the heuristic to make it consistent?



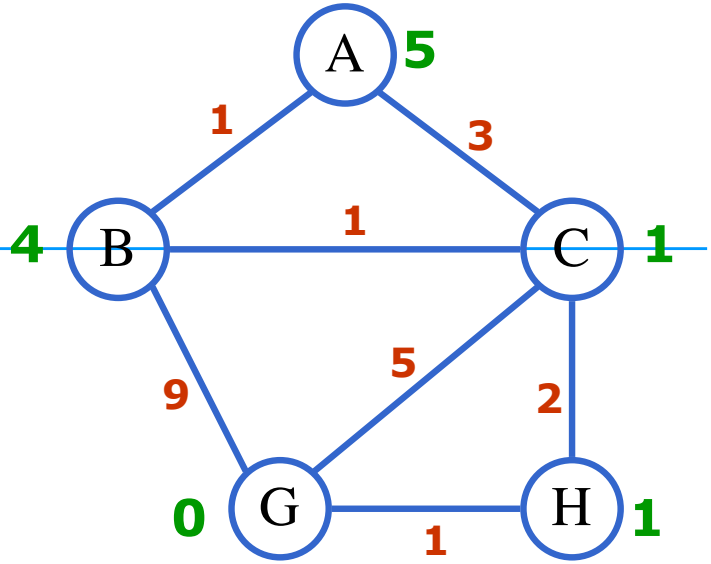
Solution
(Russel-Norvig A* on graphs)

Actual path	Cost	Heuristic	List of expanded nodes
A	0	5	



0+5 = 5

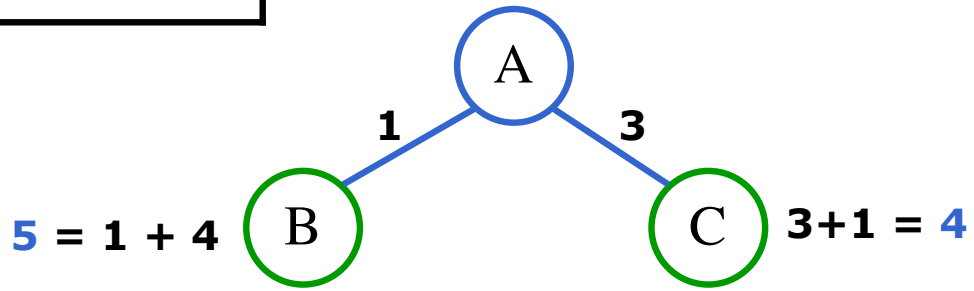
Solution
(Russel-Norvig A* on graphs)



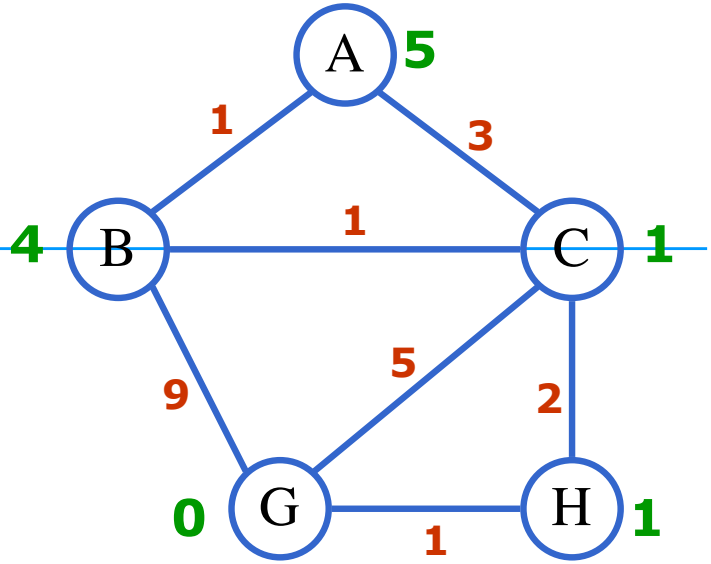
Actual path	Cost	Heuristic	List of expanded nodes
A	0	5	A

Which node to chose?

→ Node C



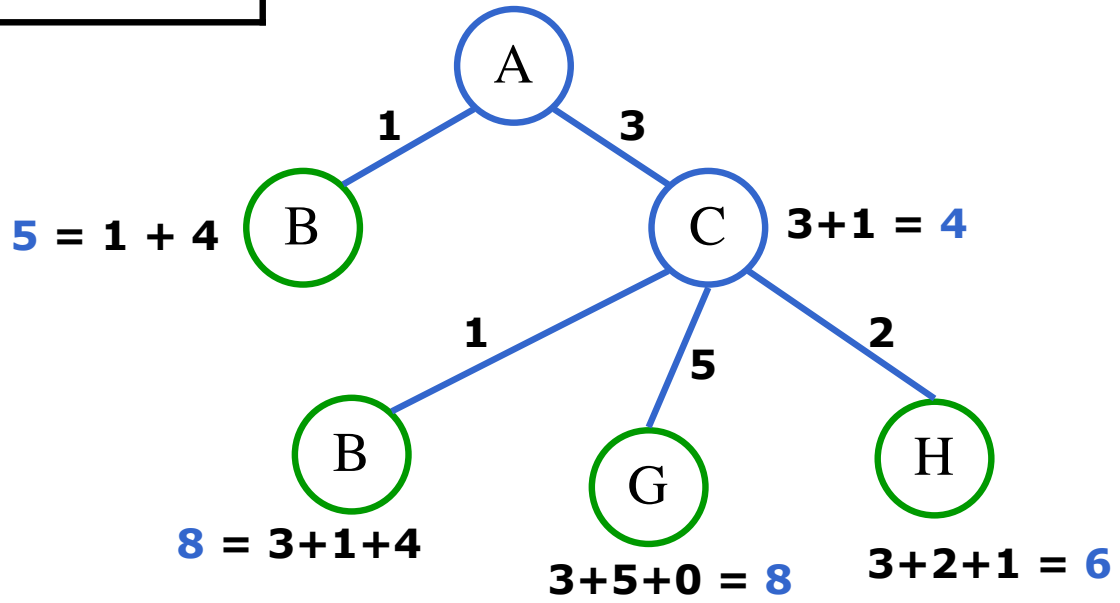
Solution
(Russel-Norvig A* on graphs)



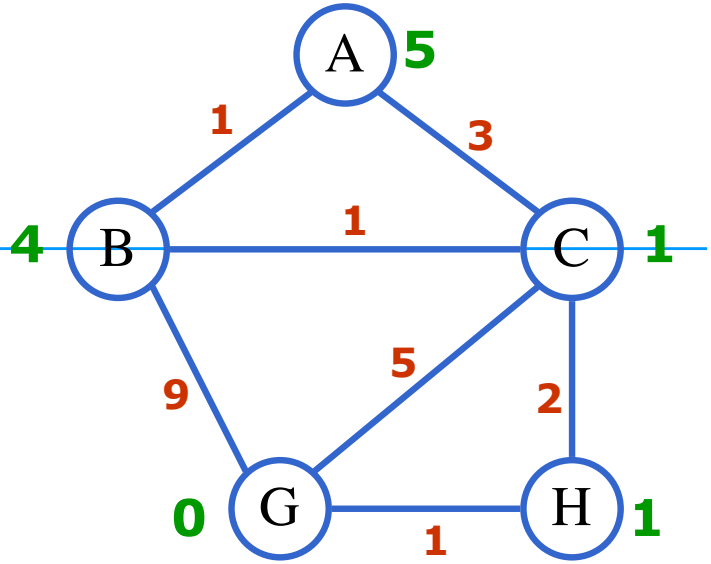
Actual path	Cost	Heuristic	List of expanded nodes
A	0	5	A
C A	3	4	C A

Which node to chose?

→ Node B

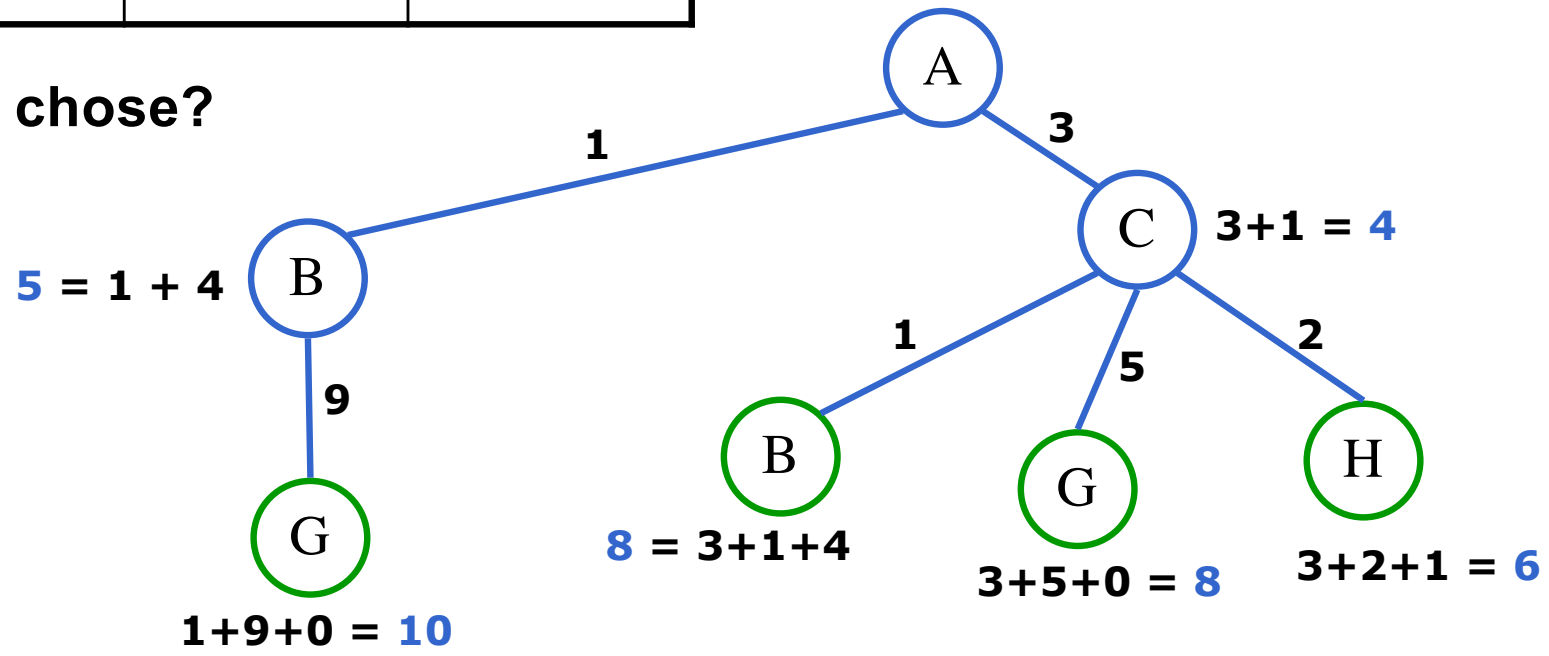


Solution (Russel-Norvig A* on graphs)



Actual path	Cost	Heuristic	List of expanded nodes
A	0	5	A
C A	3	4	C A
B A	1	5	B C A

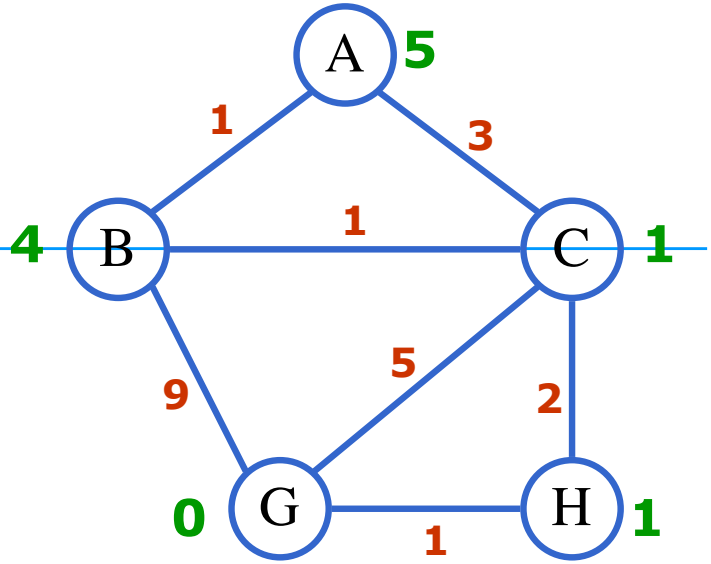
Which node to chose?



→ **Node H**

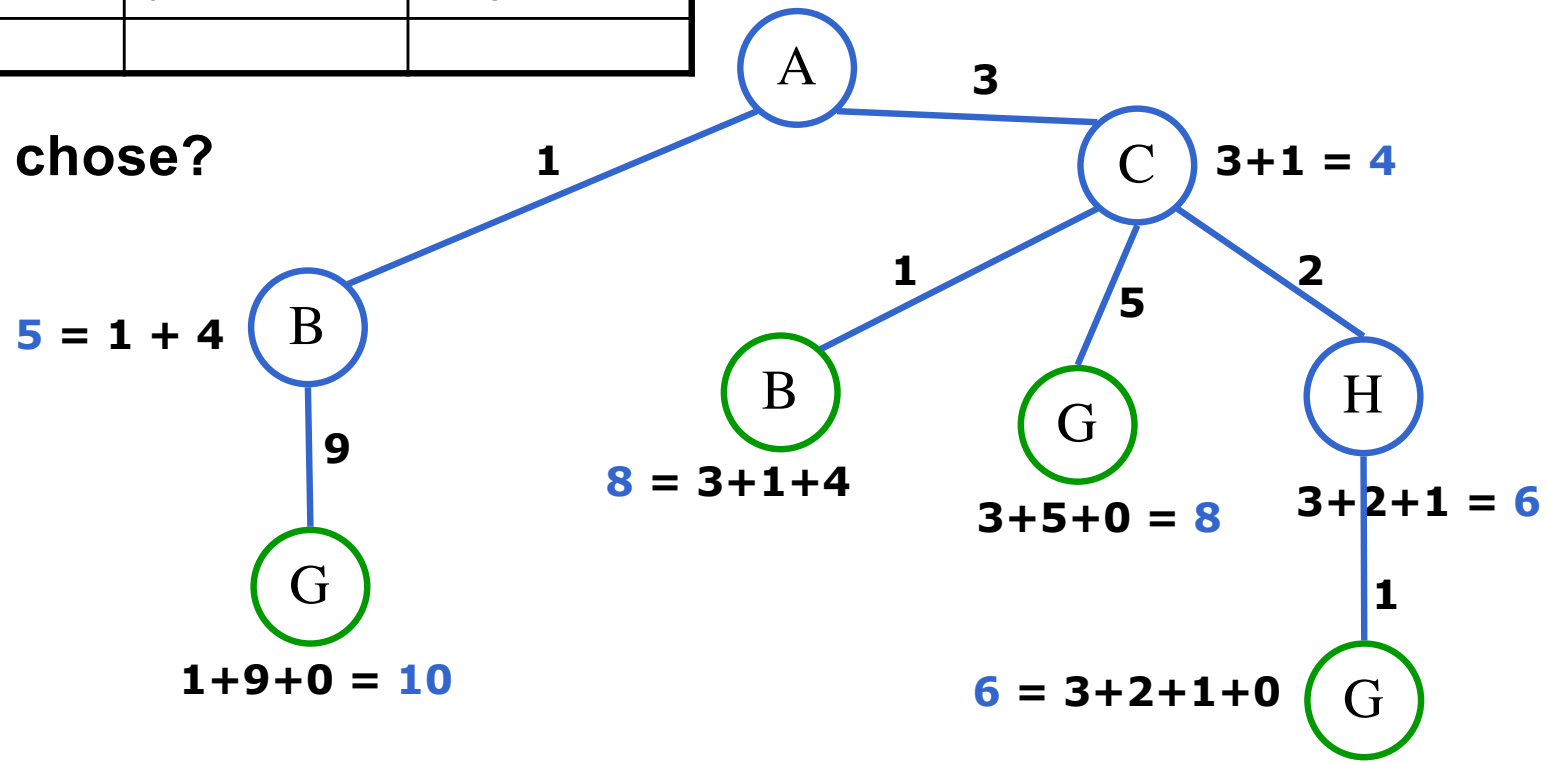
(path H C A)

Solution
(Russel-Norvig A* on graphs)



Actual path	Cost	Heuristic	List of expanded nodes
A	0	5	A
C A	3	4	C A
B A	1	5	B C A
H C A	5	6	H B C A

Which node to chose?

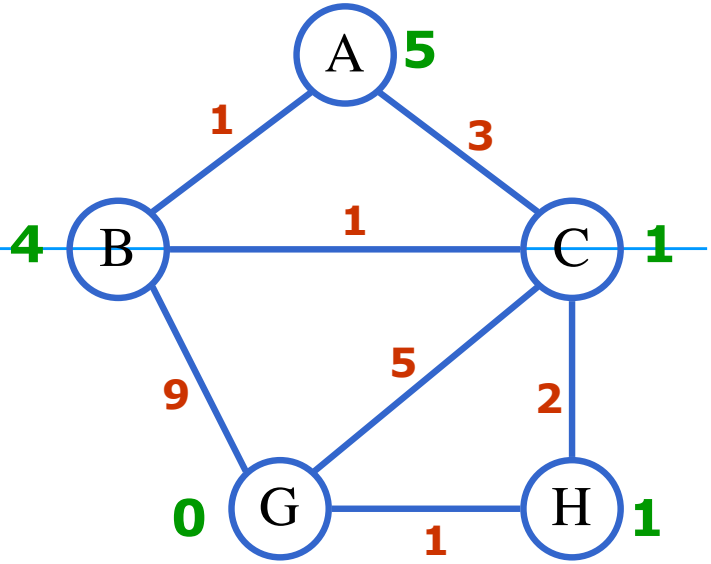


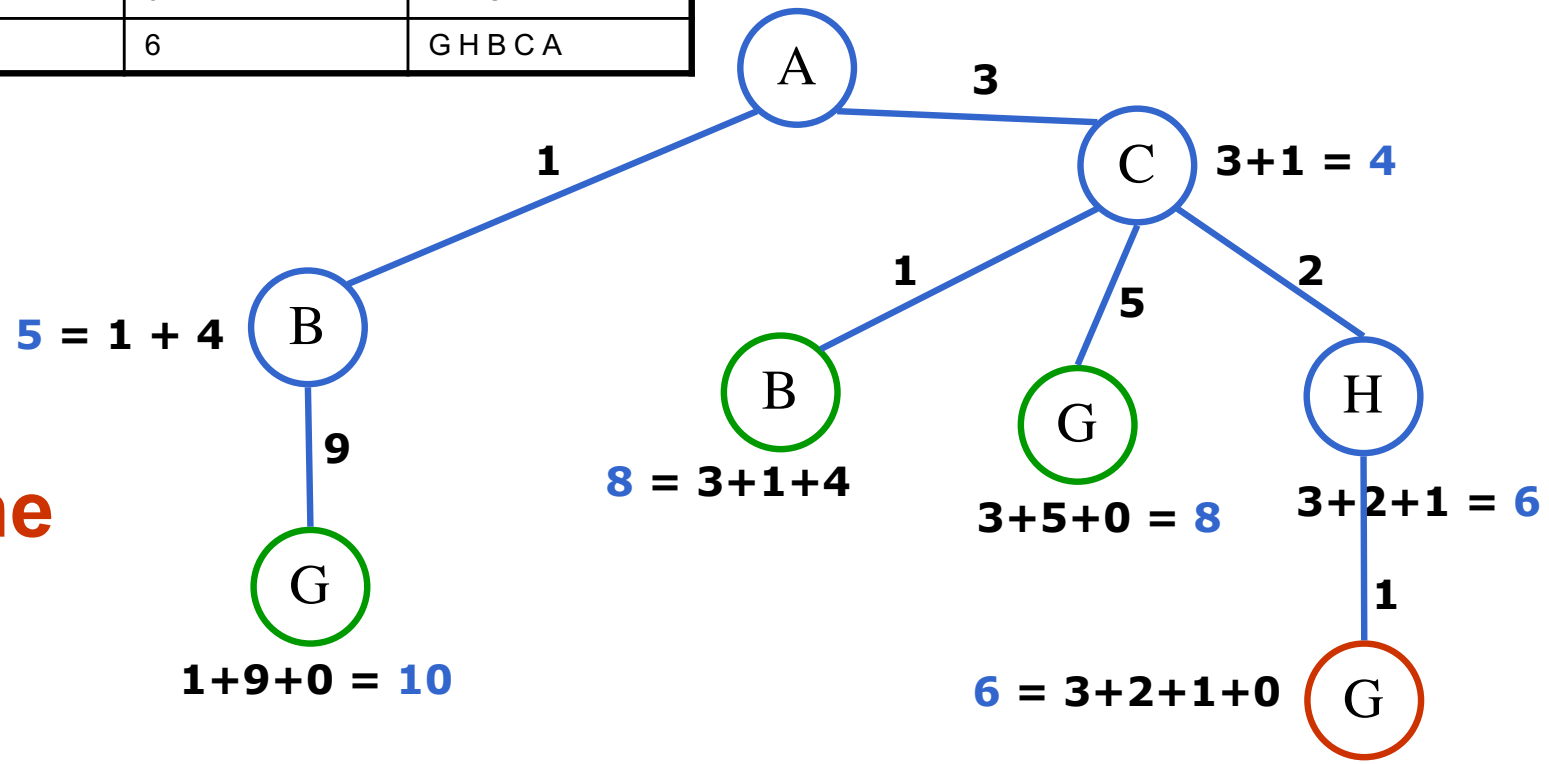
→ **node G**

(path G H C A)

Solution (Russel-Norvig A* on graphs)

Actual path	Cost	Heuristic	List of expanded nodes
A	0	5	A
C A	3	4	C A
B A	1	5	B C A
H C A	5	6	H B C A
G H C A	6	6	G H B C A

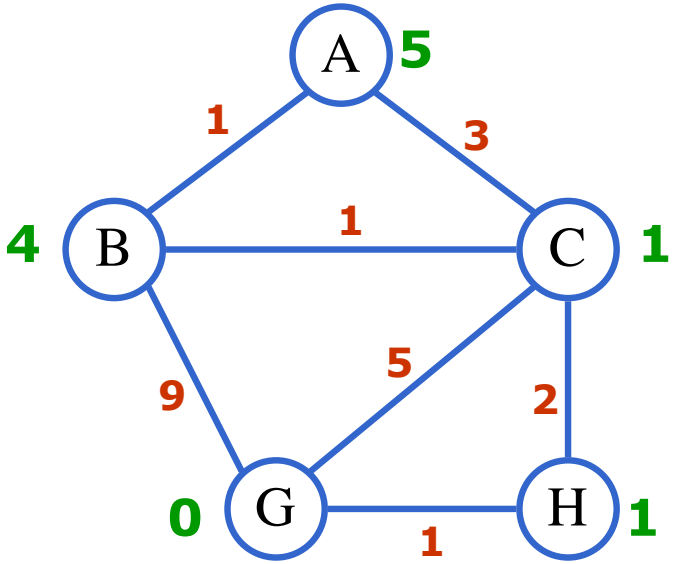




G is the GOAL!

Some observations

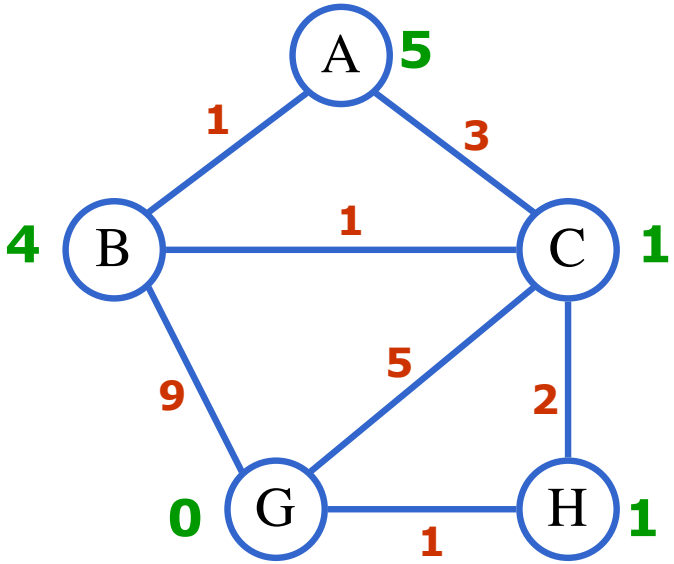
- The solution found is: ACHG, with a cost of 6.
- This is not the optimal solution. ABCHG has cost of 5
- This happens because the heuristic function is not consistent. In this case the A* algorithm for graphs does not guarantee to find the optimal solution.



- $h(n) \leq c(n, a, n') + h(n')$

Some observations

- Optimality in a A* algorithm on graphs is guaranteed if the heuristic is consistent.
- For obtaining optimality, we have two ways:
  1. Chose a consistent heuristics
  2. Modify the algorithm and updating the cost to reach a node (the g cost) each time a node is reached through a better cost path This update should then be propagated properly.



Some observations

- How to correct the heuristics?
- If we suppose that the heuristic value of C is 3, then the heuristic will be consistent and the A* algorithms for graphs will select the optimal path.

1. **La=[A], Lc=[ ]**
2. **La = [(AB,5), (AC,6)]**
   Lc = [A]
3. **La = [(ABC,5), (AC,6), (ABG,10)]**
   Lc = [A,B]
4. **La = [(ABCH,5), (AC,6), (ABCG,7), (ABG,10)]**
   Lc = [A,B,C]
5. **La = [(ABCHG,5), (AC,6), (ABCG,7), (ABG,10)]**
   Lc = [A,B,C,H]
6. **Select G and it is the goal.**