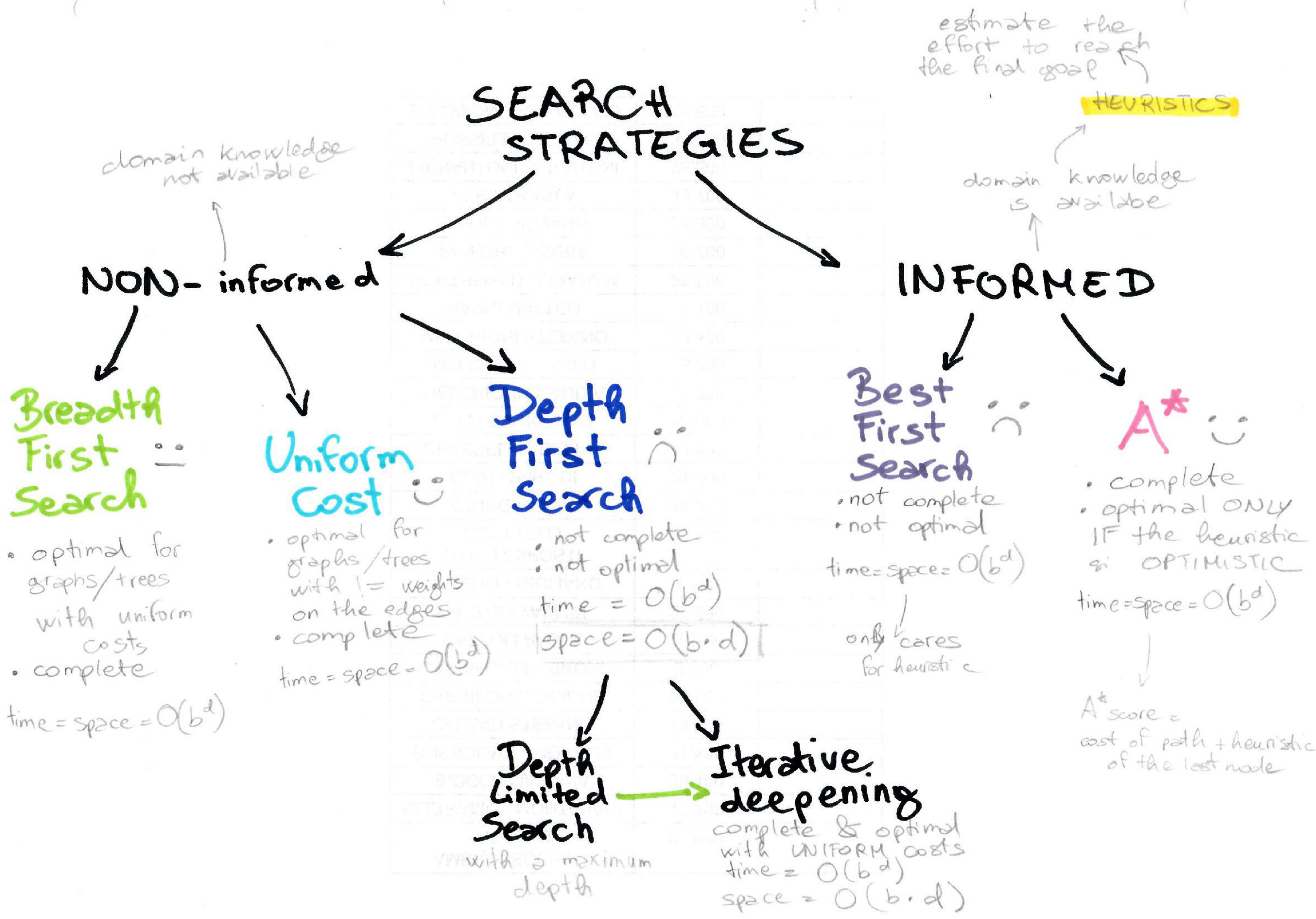


SEARCH STRATEGIES

NON- informed

domain knowledge not available

Breadth First Search

- optimal for graphs/trees with uniform costs
- complete

time = space = $O(b^d)$

Uniform Cost

- optimal for graphs/trees with != weights on the edges
- complete

time = space = $O(b^d)$

Depth First Search

- not complete
- not optimal

time = $O(b^d)$

space = $O(b \cdot d)$

Depth Limited Search

with a maximum depth

Iterative deepening

complete & optimal with UNIFORM costs

time = $O(b^d)$

space = $O(b \cdot d)$

INFORMED

domain knowledge is availbe

HEURISTICS

estimate the effort to reach the final goal

Best First Search

- not complete
- not optimal

time = space = $O(b^d)$

only cares for heuristic

A^*

- complete
- optimal ONLY IF the heuristic si OPTIMISTIC

time = space = $O(b^d)$

A^* score = cost of path + heuristic of the last node

BFS → NON-INFORMED SEARCH STRATEGY

→ domain knowledge not available

Breadth-First-Search

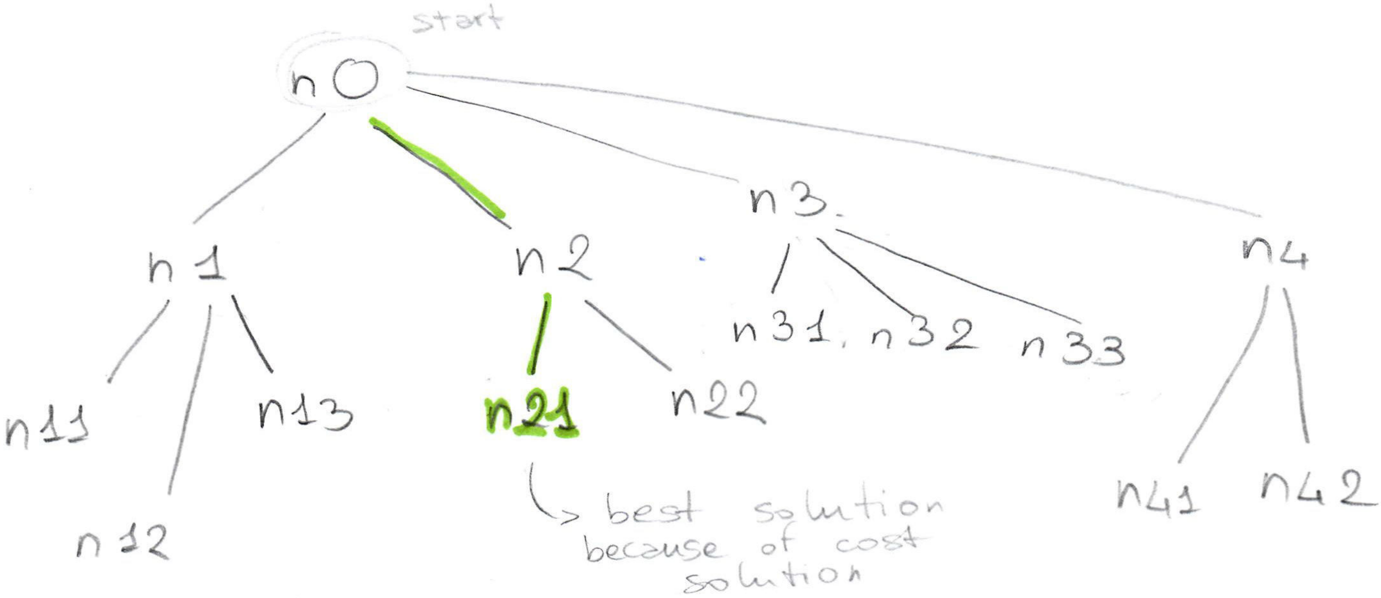
- impractical for large problems
- Completeness: Yes
- Optimality: with uniform cost → shortest path, not cheapest path !!
- Time and space complexity: $O(b^d)$

d → depth of the solution

b → branching factor → each non-leaf node has b children

→ STOPS at the first solution: shortest path !!

Ex 1.



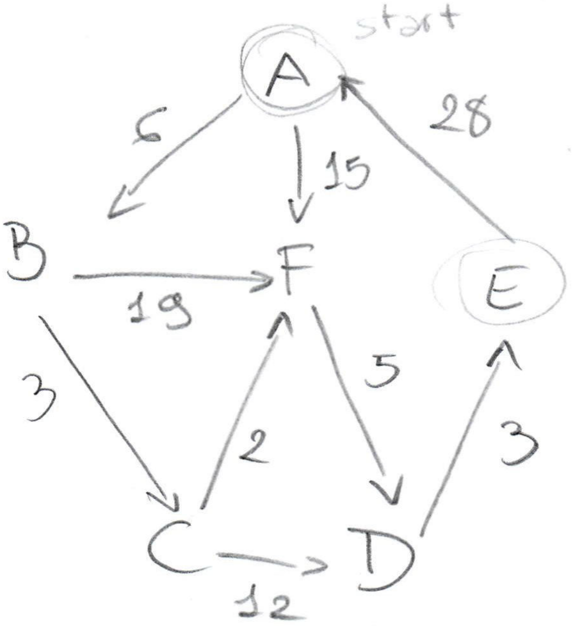
Solve with BFS, the goal is n21

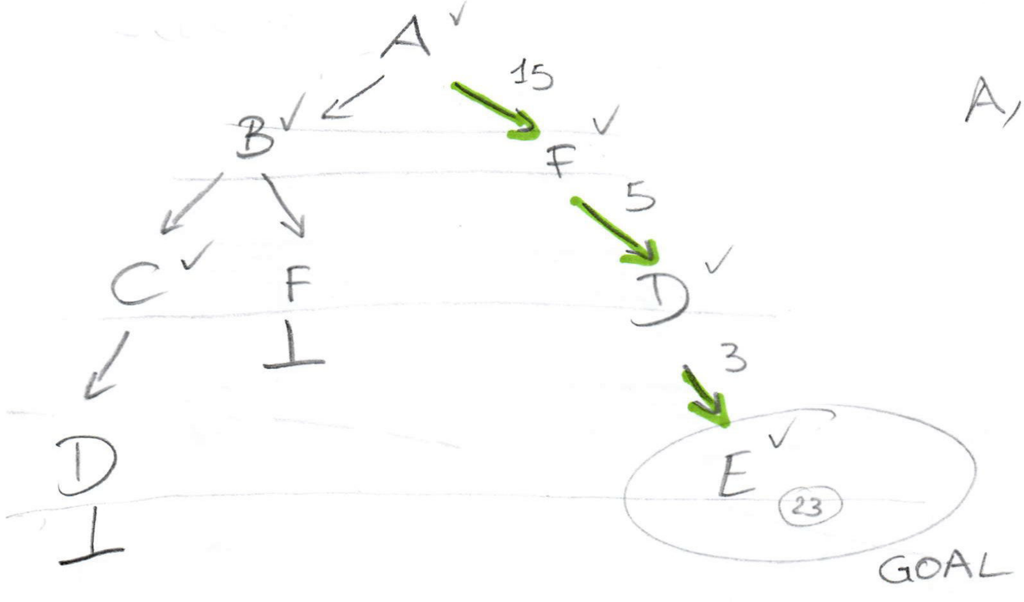
expanded nodes:

- n0
- n1, n2, n3, n4
- n2, n3, n4, n11, n12, n13
- n3, n4, n11, n12, n13, n21, n22
- n4, n11, n12, n13, n21, n22, n31, n32, n33
- ~~n11~~, n12, n13, n21, n22, n31, n32, n33, n41, n42
- ~~n12~~, n13, n21, n22, n31, n32, n33, n41, n42
- ~~n13~~ n21, n22, n31, n32, n33, n41, n42
- **n21**, n22, n31, n32, n33, n41, n42

Ex 2.

Solve with BFS, the goal is E





Visited

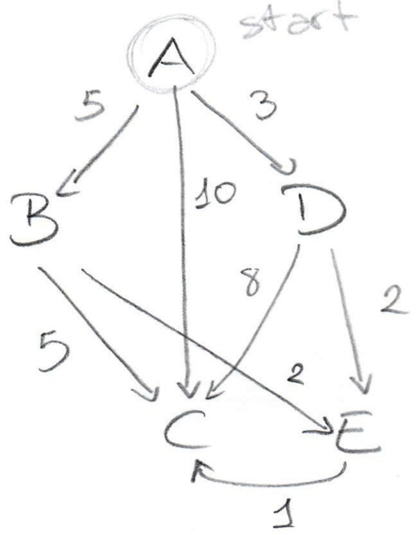
A, B, F, C, D, E

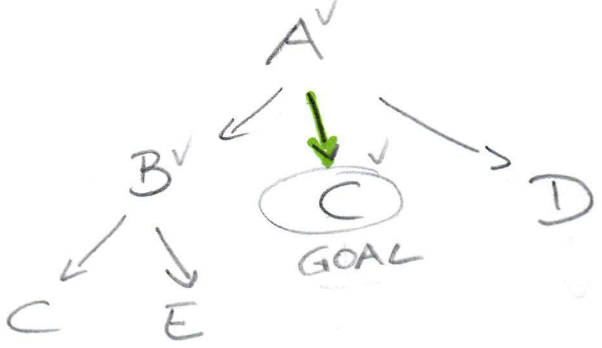
solution cost = path cost = 23

→ might not be the best solution as the edges DO NOT have uniform costs

Ex 3.

Solve with BFS, the goal is C





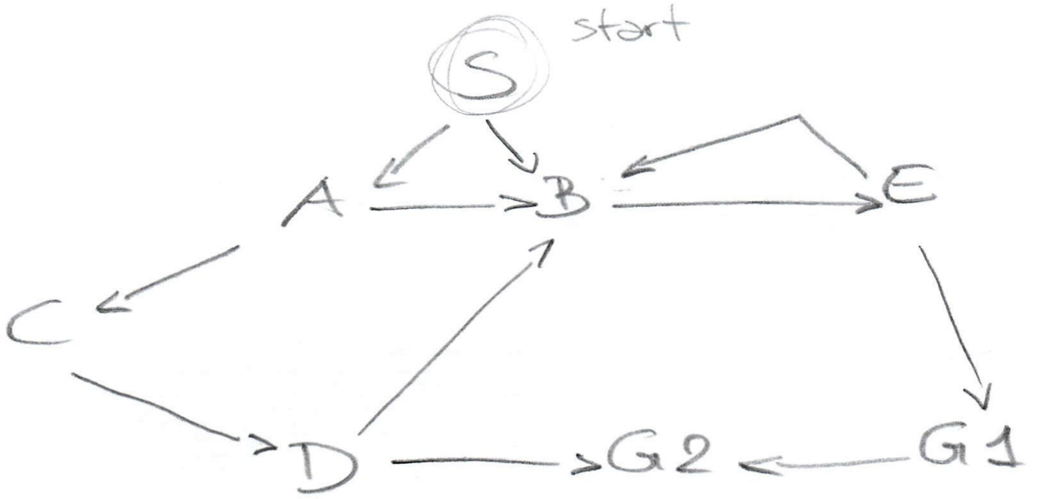
Visited
A, B, C

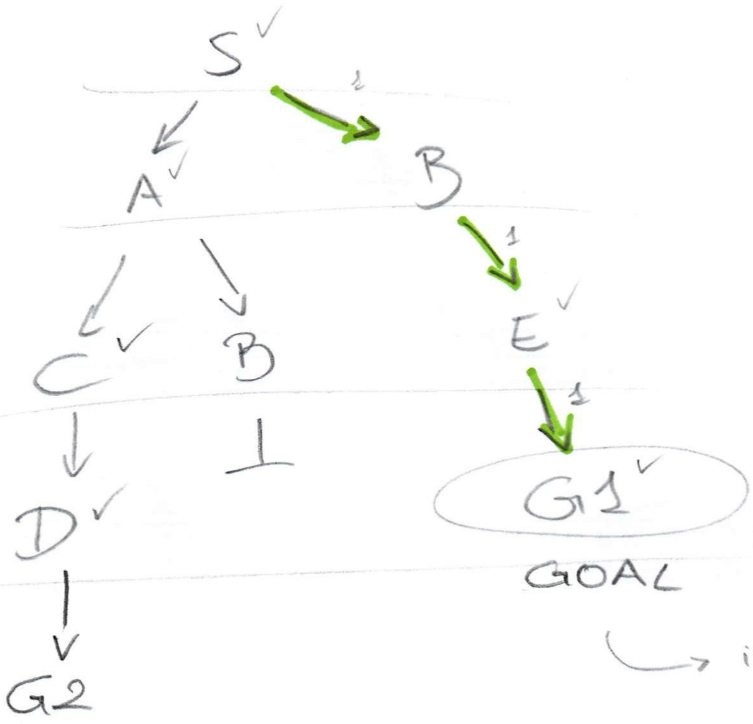
solution cost = path cost = 10

might not be the best solution as the edges DO NOT heve uniform costs

Ex 4.

Find the solution with BFS, either G1 or G2





Visited
S, A, B, C, E,
D,

↳ is the best solution, because costs are uniform

solution cost = path cost = 3

Uniform-cost search

Completeness	Yes	
Optimality	Yes	
Time and space complexity	$O(b^d)$	

→ depth of the solution

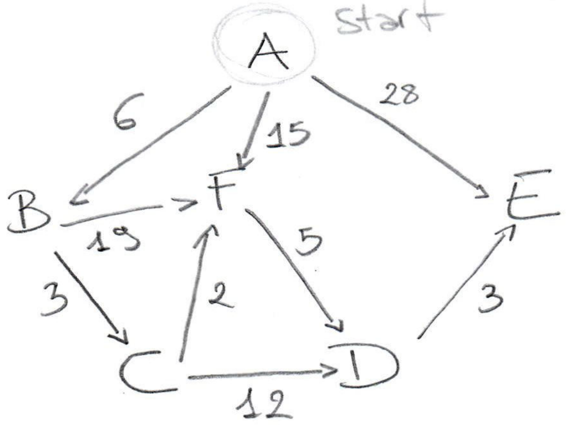
branching factor

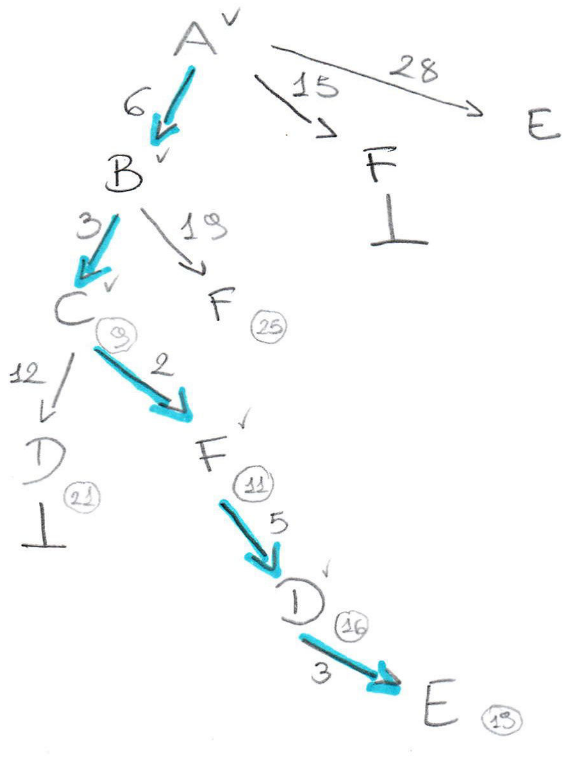
→ first are visited the path with lower cost, following the BFS logic

→ based on Dijkstra Algorithm, in order to find the cheapest path

Ex 1.

Solve with UNIFORM COST, E is the goal





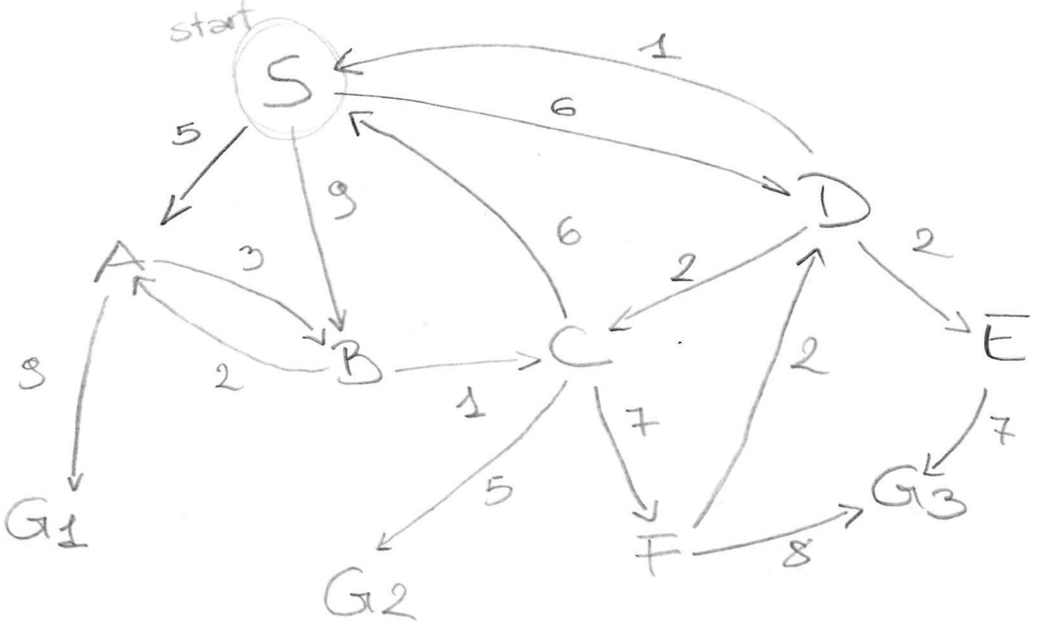
Visited

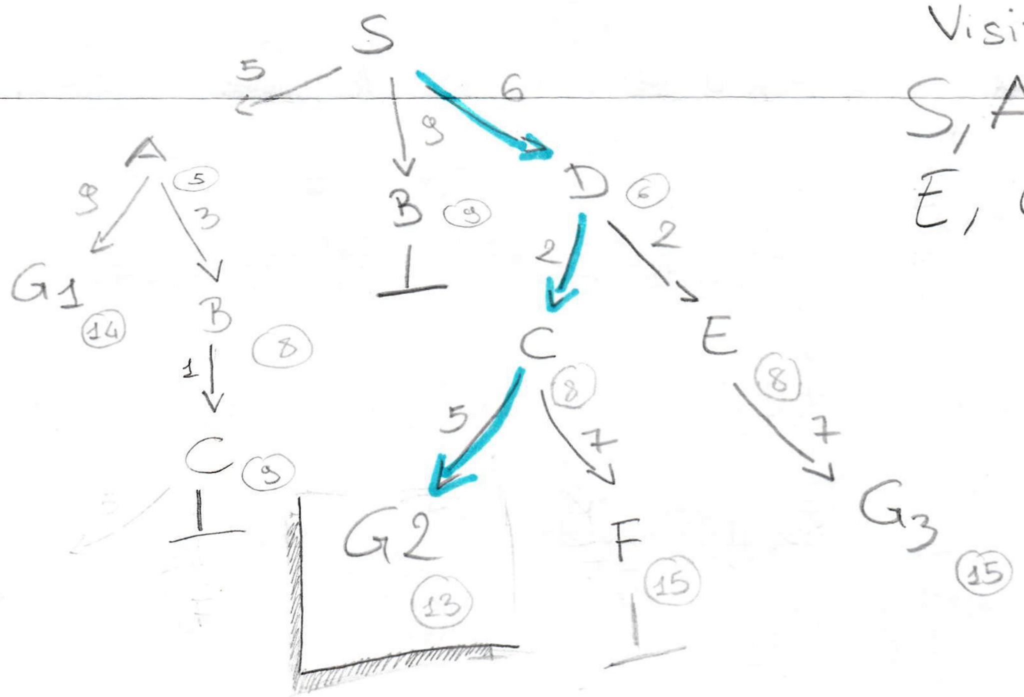
A, B, C, F, D, E

$\boxed{\text{solution cost} = 19}$

Ex 2.

Solve with <u>uniform</u> cost search, the goal is either G1, G2 or G3





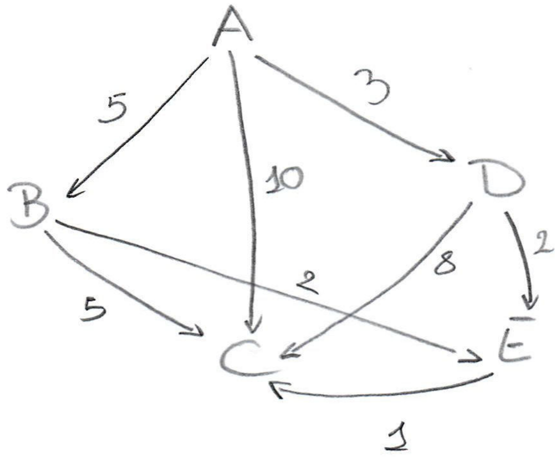
Visited nodes

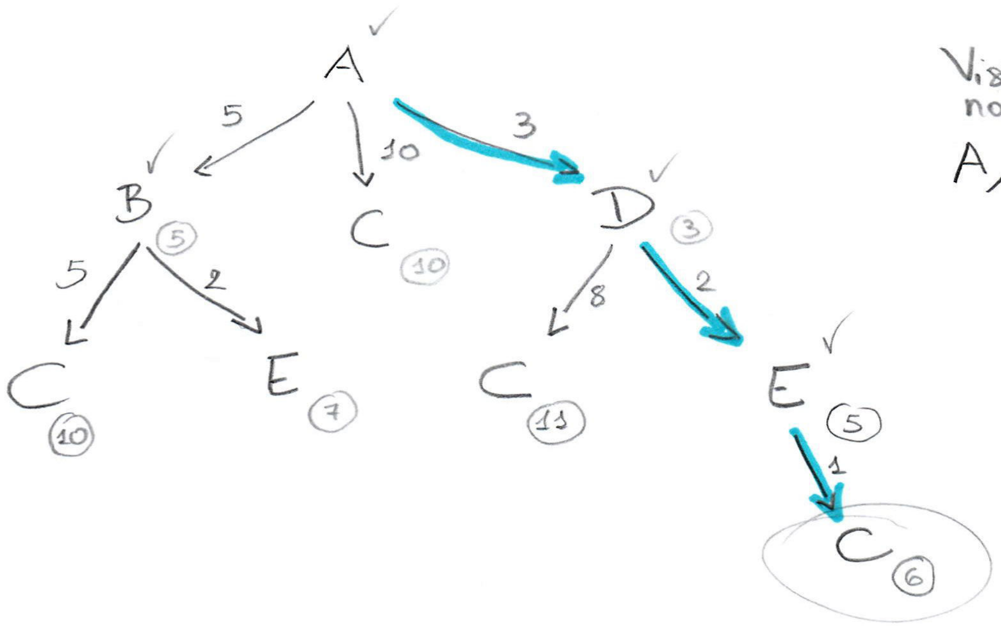
S, A, D, B, C, E, G_2

solution cost = 13

Ex 3.

Sdve with uniform cost search, the GOAL is C, start from A





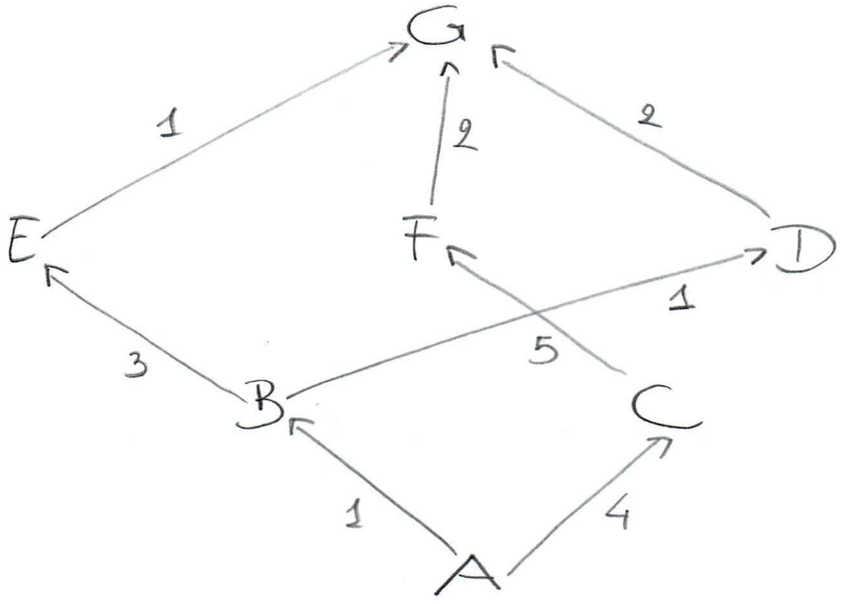
Visited nodes

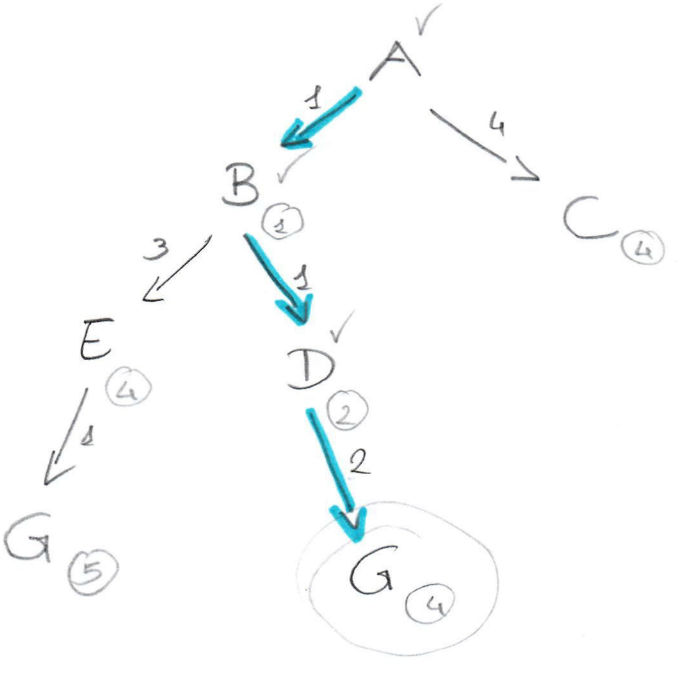
A, D, B, E, C

solution cost = 6

Ex 4.

Solve with uniform cost search:
A is the START point, G is the GOAL





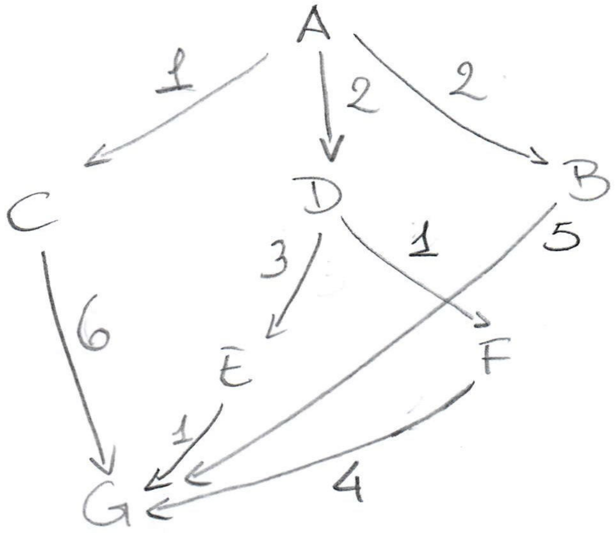
Visited nodes

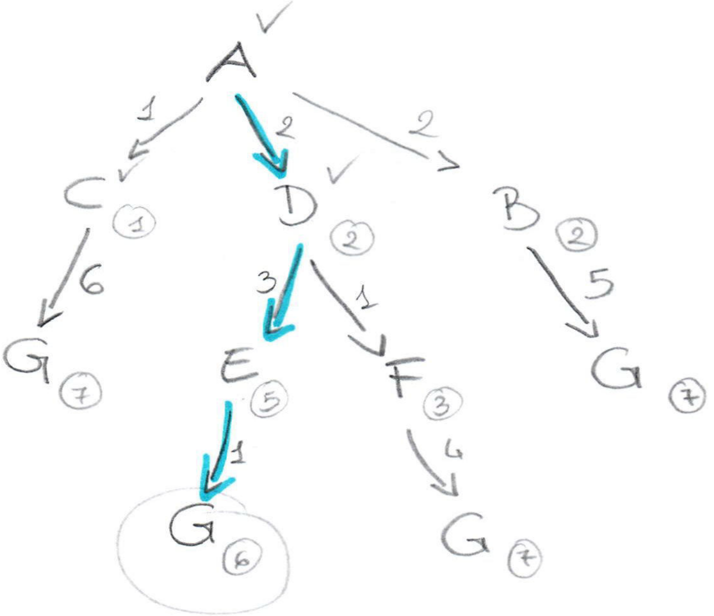
A, B, D, (E), G

solution cost = 4

Ex 5.

Solve with uniform cost
A is start and G is the GOAL





Visited nodes
A, C, D, B, E, F, G

solution cost = 6

DFS

OFTEN

Depth - First - Search

- COMPLETENESS : No -> loops
- OPTIMALITY : No
- TIME COMPLEXITY : $O(b^d)$
- SPACE COMPLEXITY : $O(b\ d)$

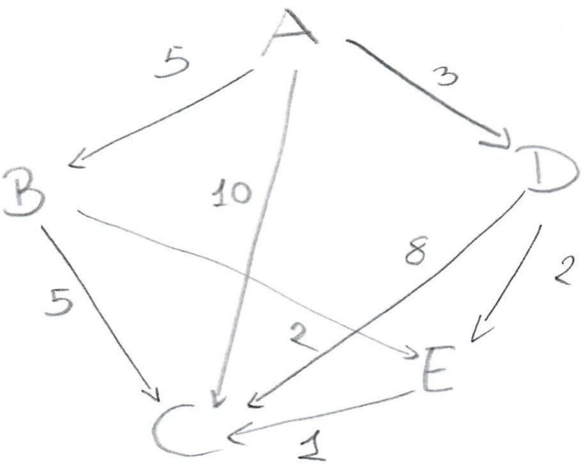
branching factor

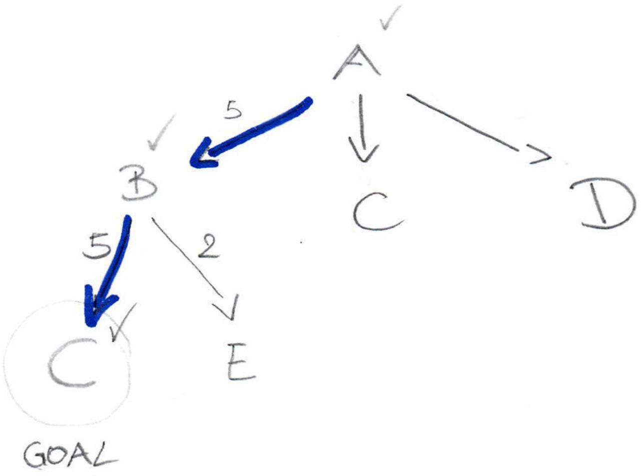
maximum depth

-> very quickly, less use of memory space, not an optimal plan, too many actions

Ex 1.

Solve with DFS, A is START and C the GOAL





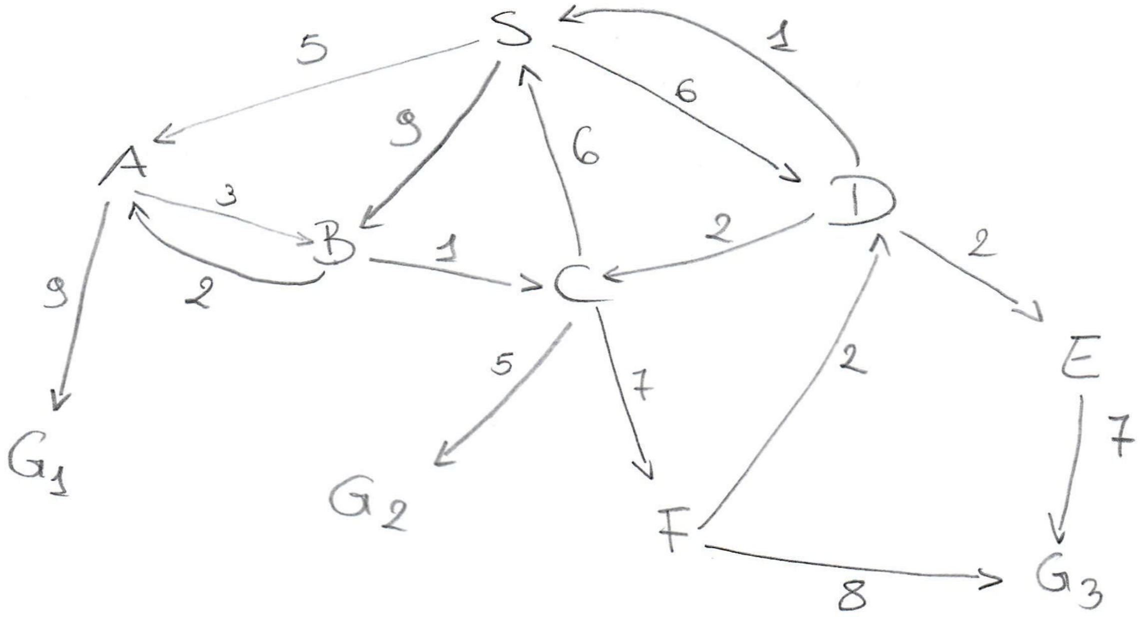
Visited nodes

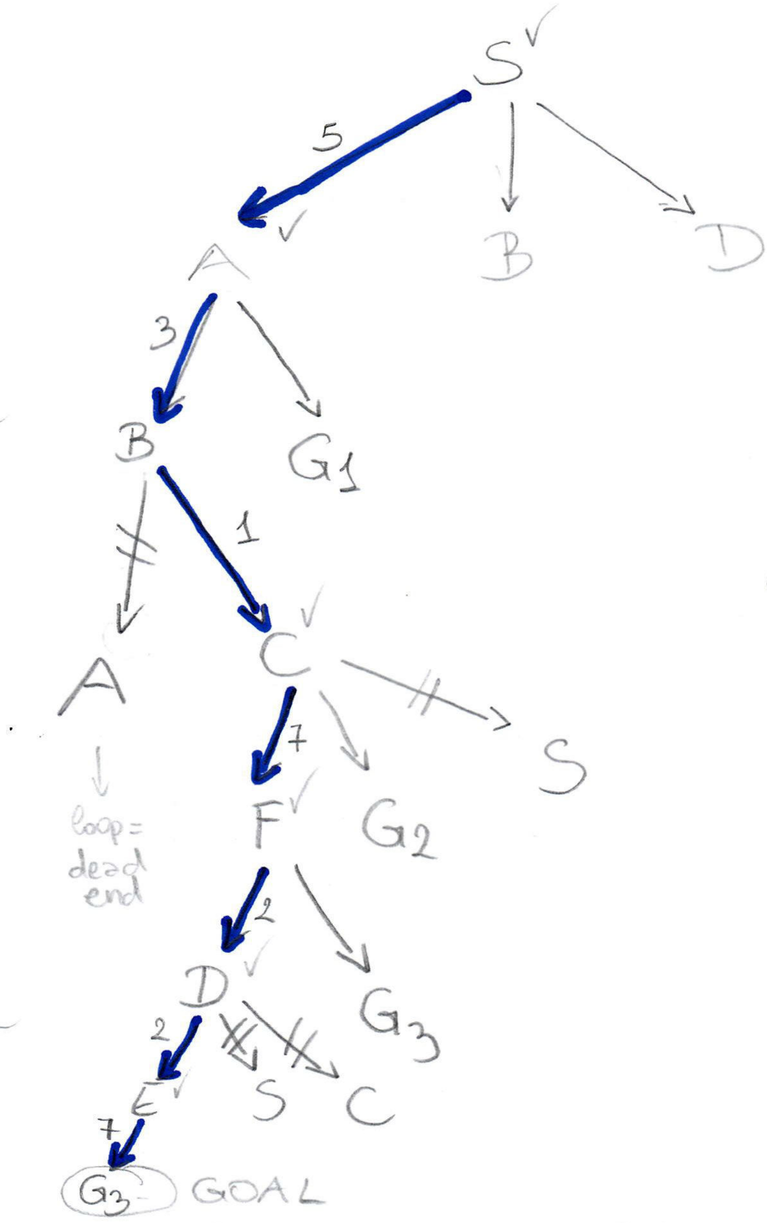
A, B, C

solution cost = 10

Ex 2.

Solve with DFS, S is start, either G_1, G_2 or G_3 is the goal





Visited nodes

S, A, B, C, F, D

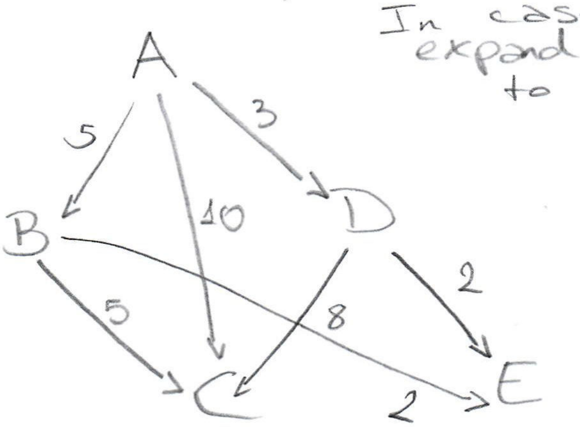
to avoid loops!!

G_3 solution cost = G_3

Ex 3.

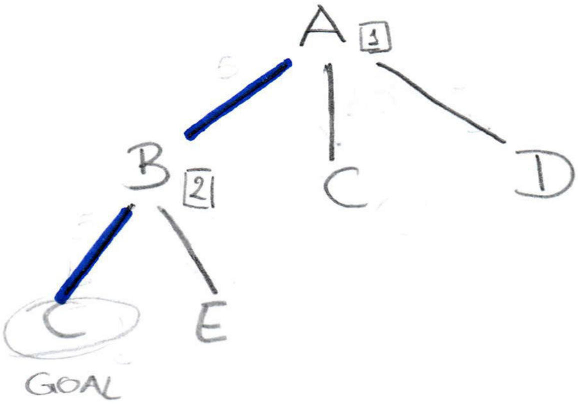
Solve with DFS, A is the START and C the goal

In case of non-determinism expand the nodes according to alphabetical order



Visited nodes

A, B, C

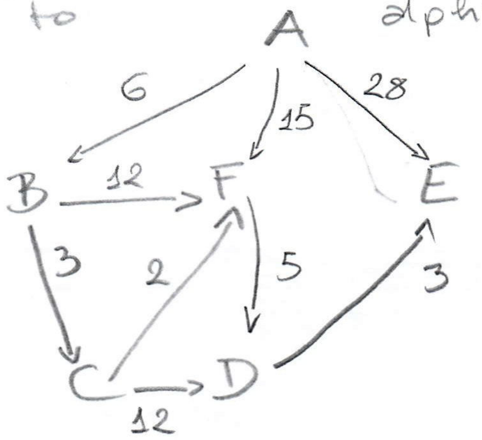


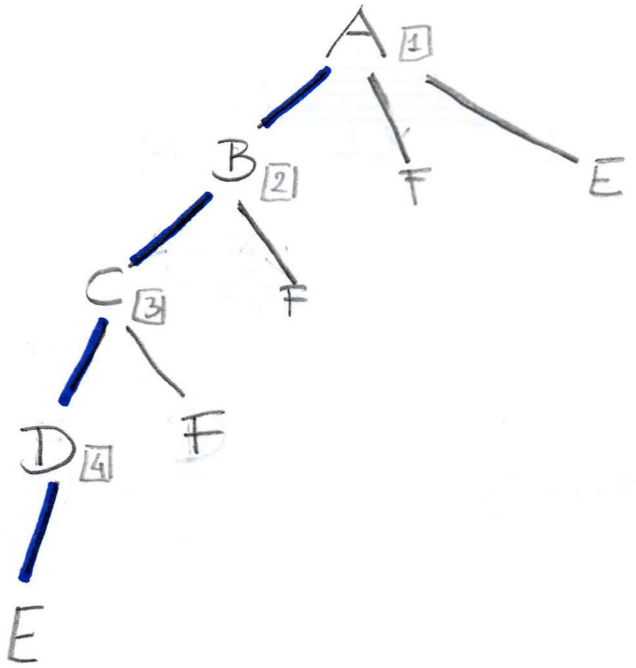
Cost of the path found (in blue), ABC equal to 10

checked with professor's solution!!

Ex 4.

Solve with DFS: A is the start and E the end
In case of non-determism expand the nodes according to alphabetical order





Visited nodes

A, B, C, D, E

1 = order of visit

The cost of the solution path in blue, ABCDE is 24

✓ checked with professor solution

Best- First search

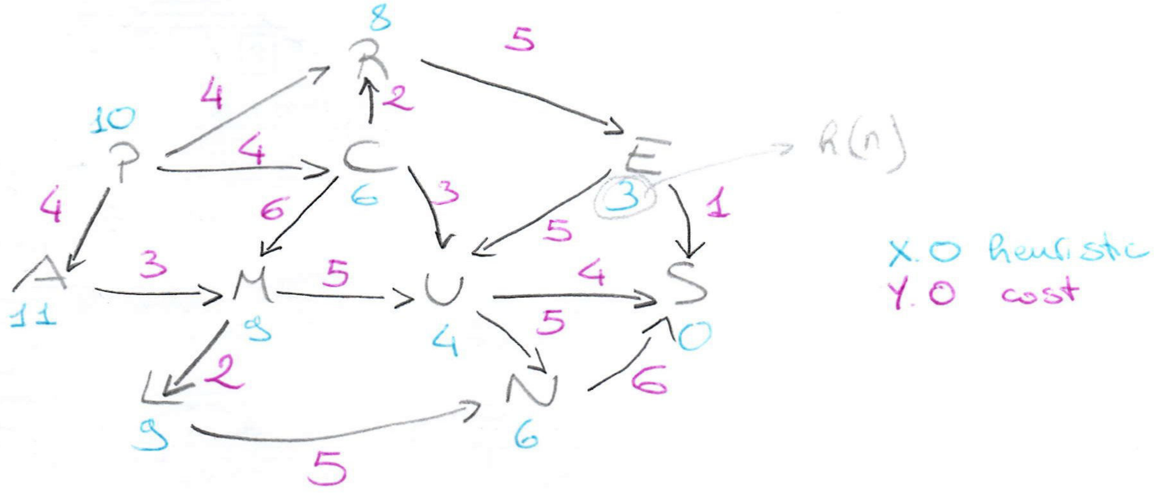
OFTEN

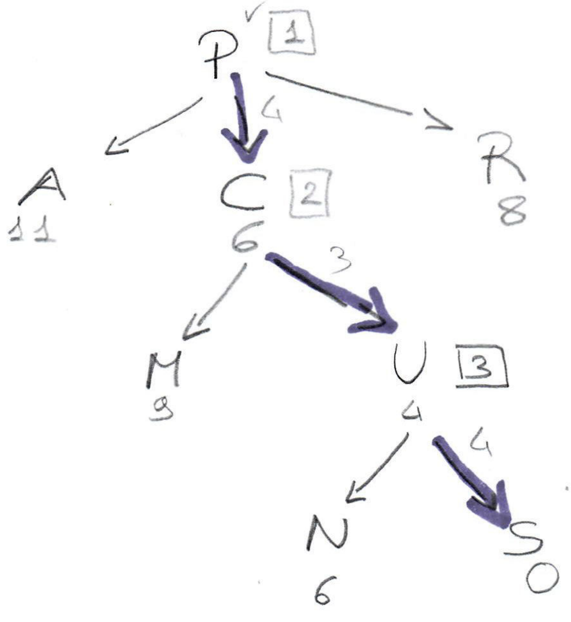
- completeness : NO
- optimality : NO
- time complexity : $O(b^d)$
- space complexity : $O(b^d)$ → solution depth; b → branching factor

→ choosing the next node to explore based on the value of h(n) which indicates how near the solution is

Ex 1.

Solve with greedy Best-First Search
P is the START and S is the goal



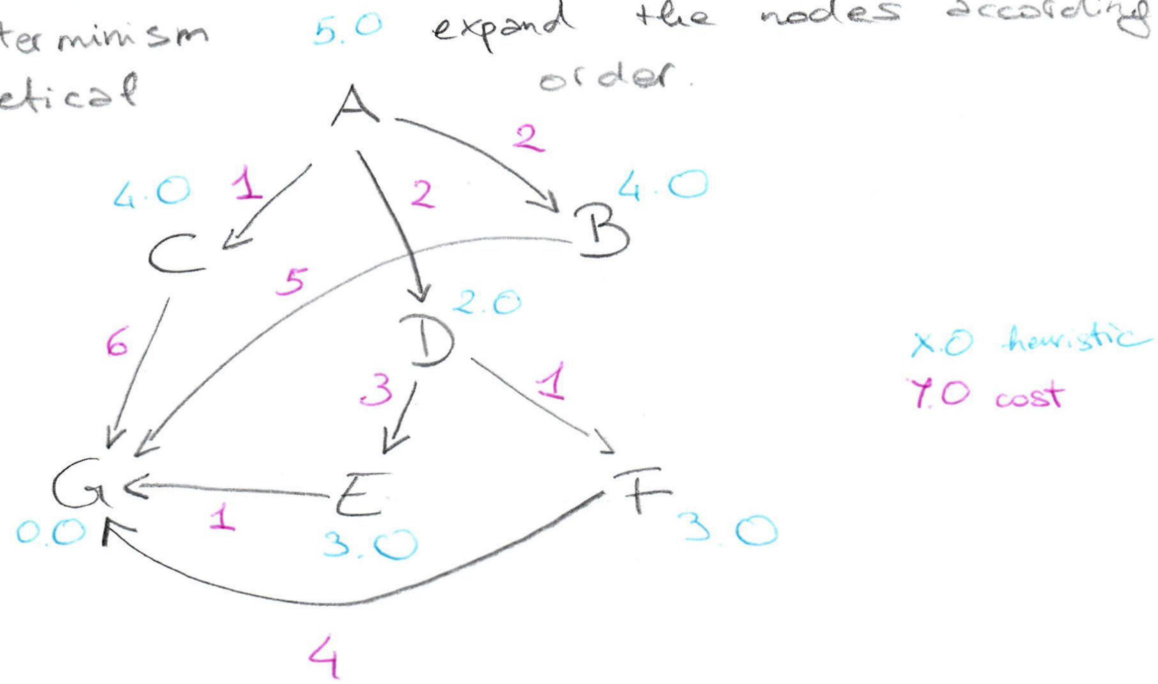


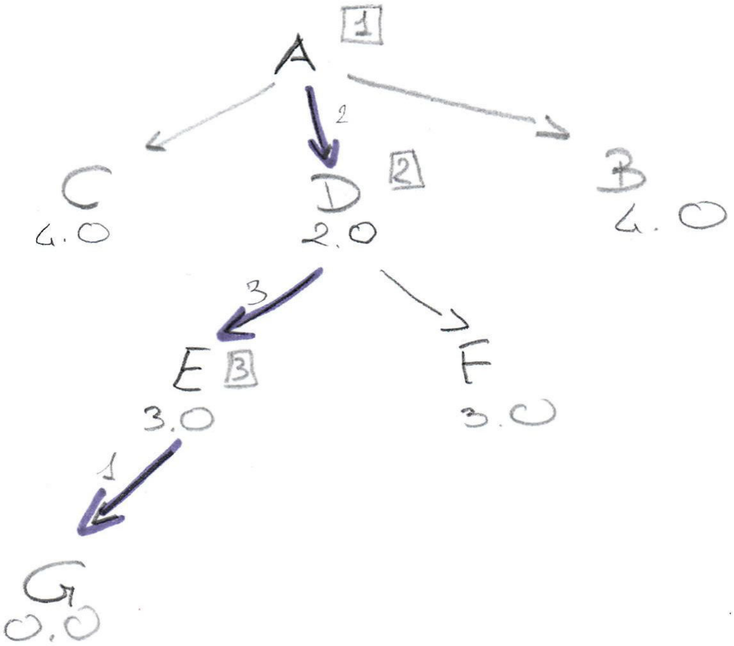
Visited nodes
P, C, U, N, S

The cost of the solution path in purple, PCUS is 4+3+4 = 11

Ex 2.

Solve with Best-First-Search, A is the start and G the GOAL; In case of non-determinism expand the nodes according to alphabetical order.



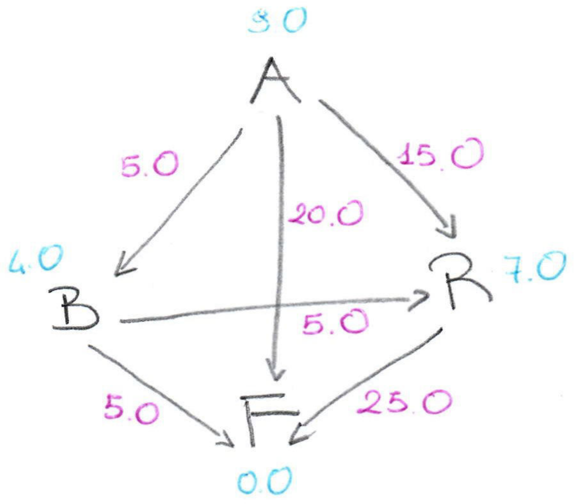


visited nodes
A, D, E, G

The cost of the solution path ADEG, in purple, is 2+3+1 = 6

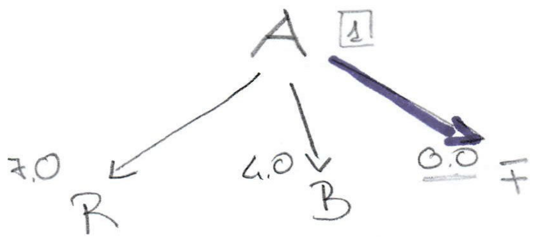
Ex 3.

Solve with Best-First-Search, A is the start and F the GOAL



X.0 stima heuristic

Y.0 costo



visited nodes
A, F

The cost of the solution path AF (in purple) is 20

✓ solution checked with professor

A*

OFTEN

A star

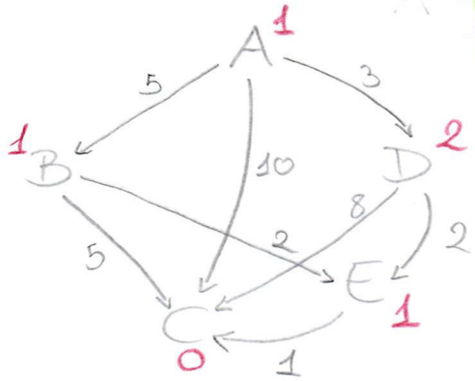
- completeness: Yes
- optimality: only if the heuristic is optimist
- time complexity: $O(b^d)$
- space complexity: $O(b^d)$

A^* score = cost of path + heuristic of the last node

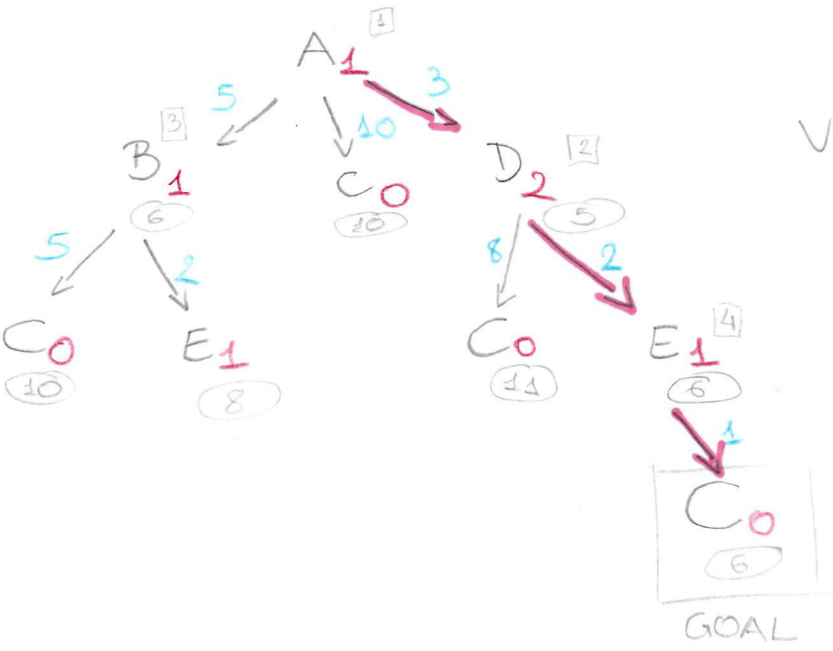
Ex 1.

Solve with A^*, A is the start and C the end. In case of non-determinism, choose the nodes to be expanded according to the alphabetical order.

Is the heuristic h defined in this way admissible?



The heuristic is admissible because it never overestimates the cost of reaching the goal, regardless of which node of the graph you consider as starting point



Visited nodes

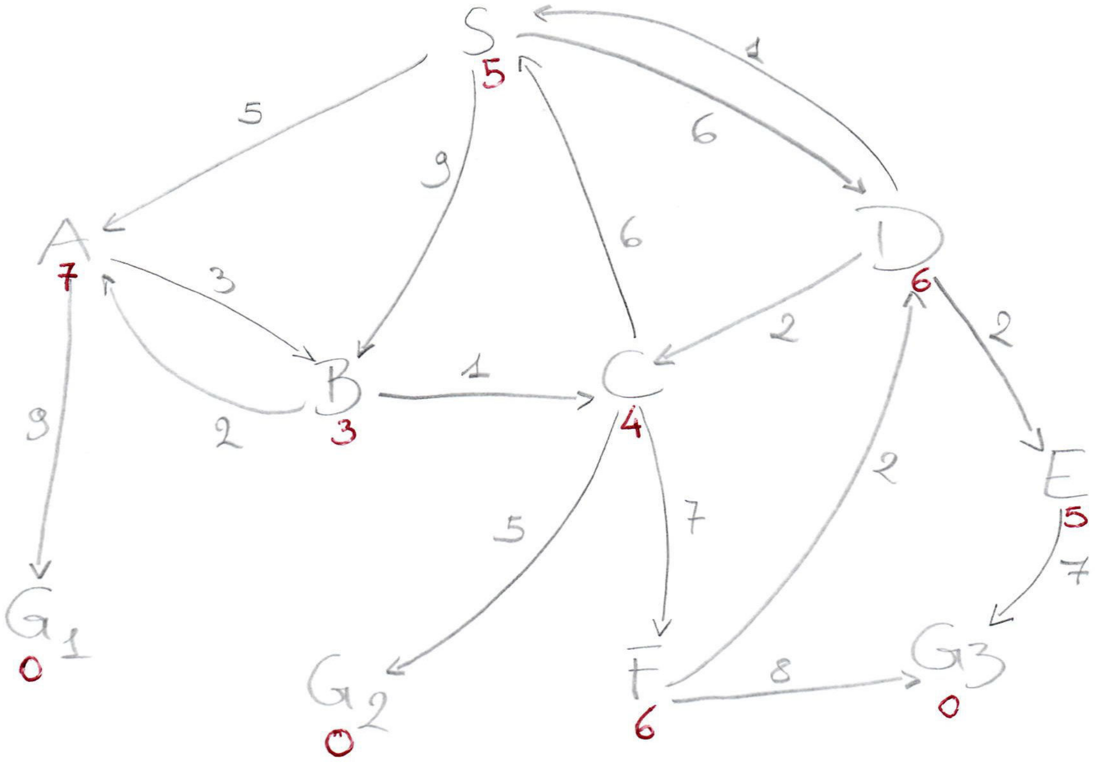
A (1), D (5), B (6), E (6)

solution cost = 6

✓ solution checked with professor

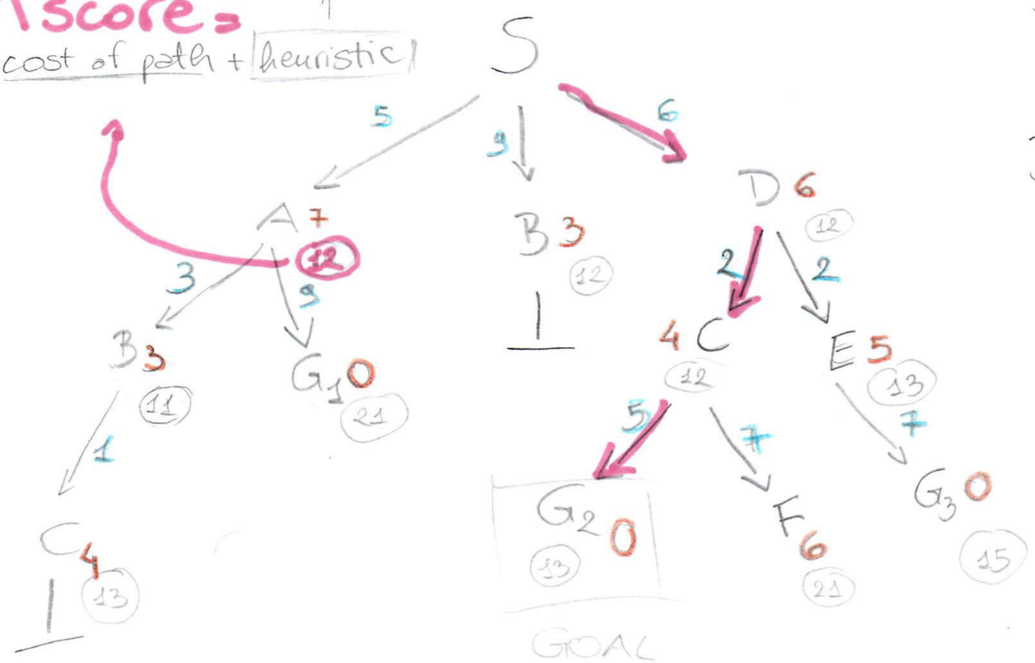
Ex 2.

Solve with A*, start is S and the goal is either G_1, G_2 or G_3



A* score = cost of path + heuristic

ONLY of the last node



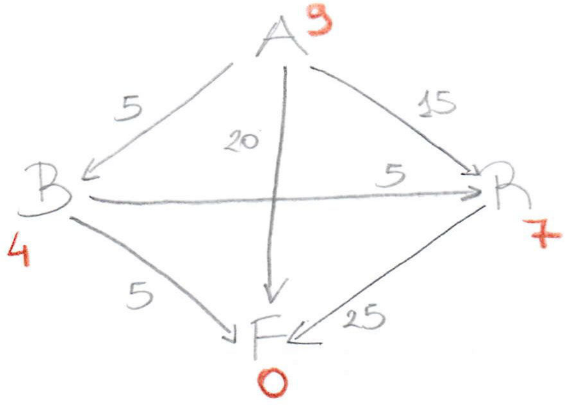
Visited
S (5), A (12), B (11), D (12), C (12), E (13)

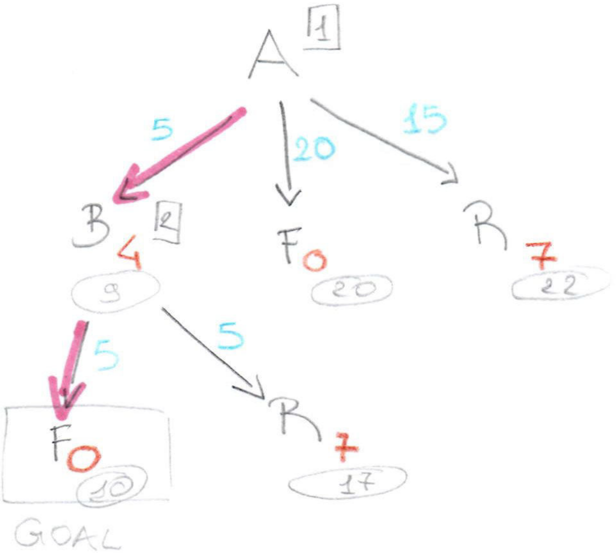
path cost
heuristic

Solution cost = 13

Ex 3.

Which is the cost of the path found by A*?
A is the START and F the GOAL





Visited nodes
A(9), B(9)

solution cost = 10

Is the heuristic admissible?

Yes, because it never overestimates the cost of reaching the goal, regardless of which node of the graph you consider as starting point!

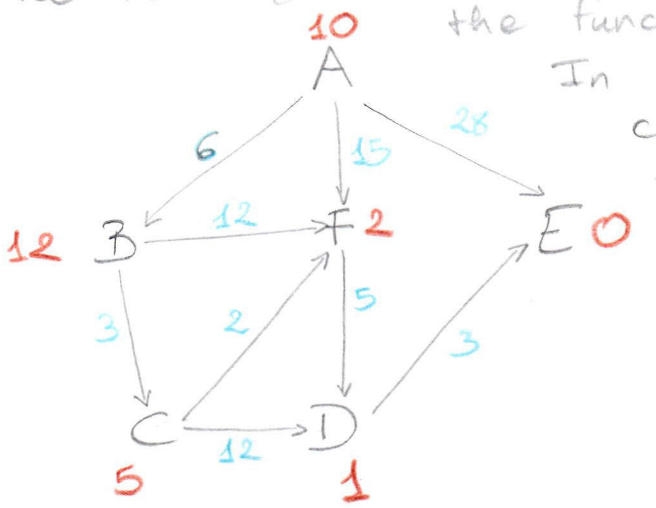
✓ solution checked with prof

Ex 4.

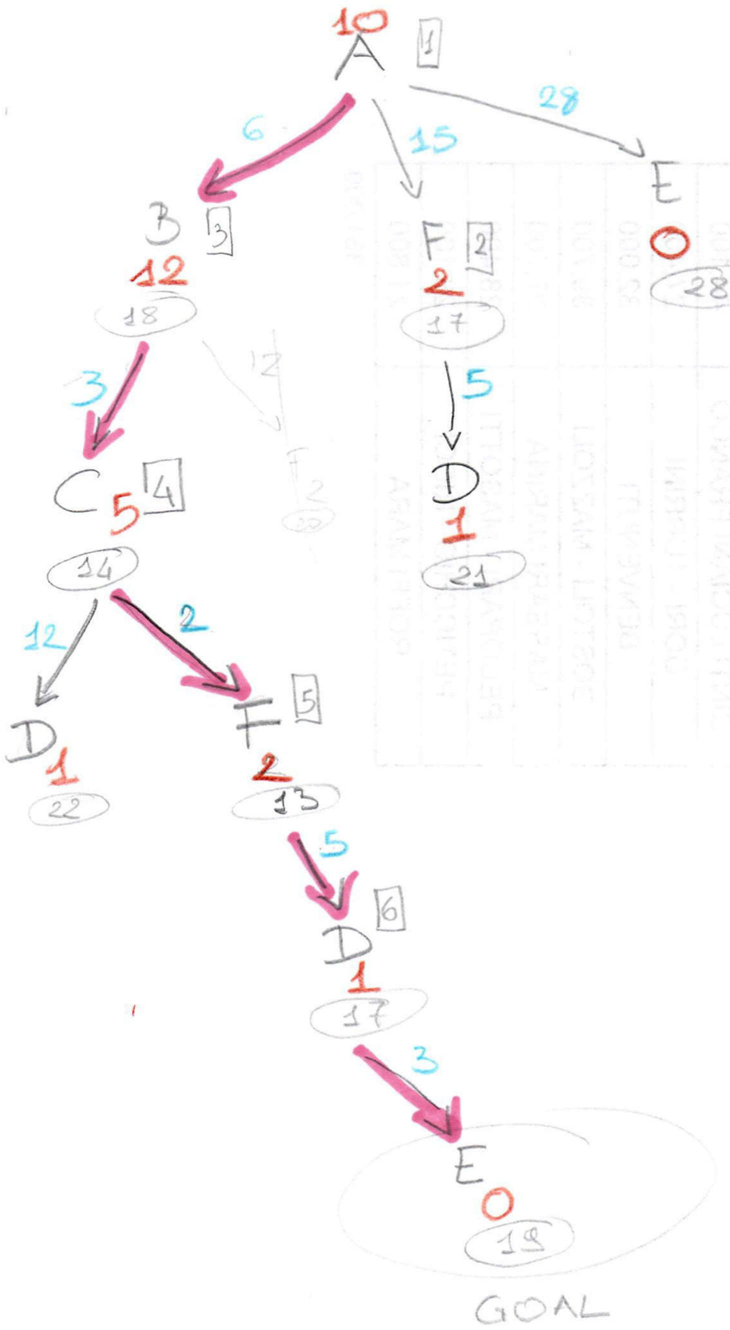
Apply search A*, and draw the developed search tree indicating the expansion order and the value of the function f(n) for each node n.

In case of non-determinism, chose the nodes to expand according to alphabetical order

What is the produced solution and its cost?



◯ = f(n)



Visited nodes

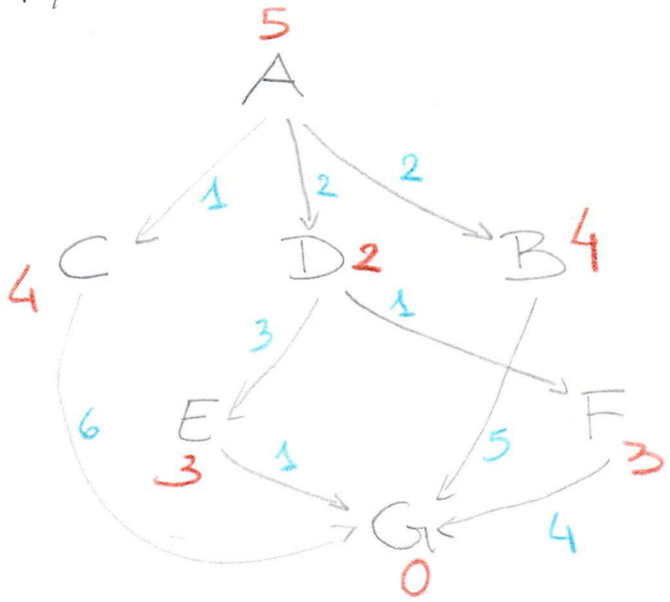
A(10), F(17), B(3), C(14), F (13),

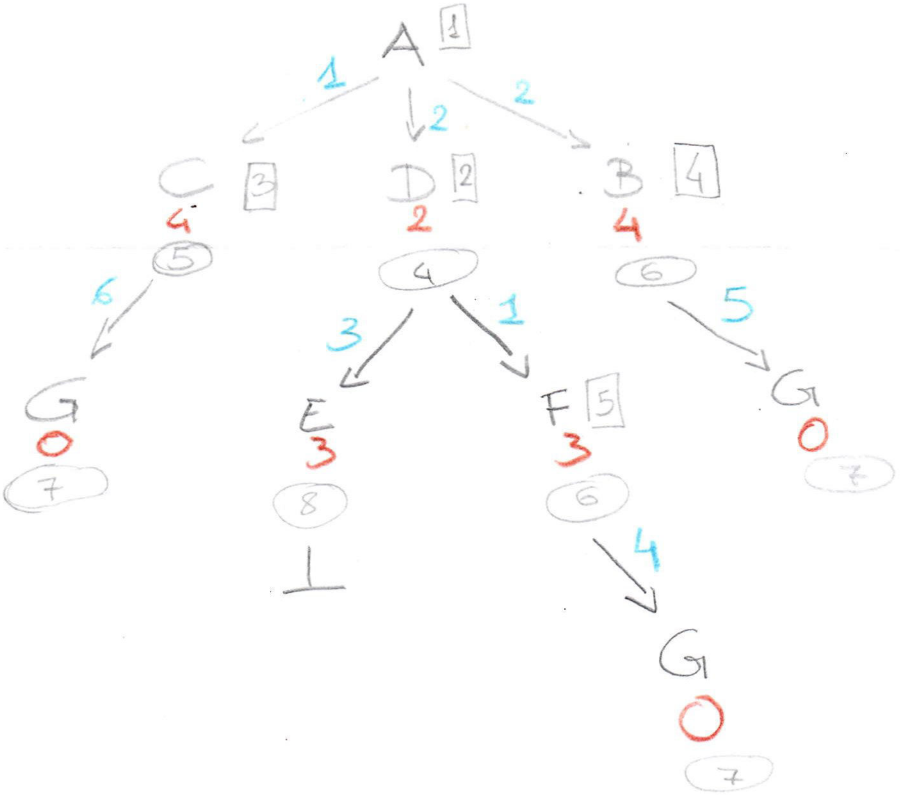
Solution cost = 19

solution checked with professor

Ex 5.

Apply A*, A is the START and G the GOAL





Visited nodes

A(5), D(4), B(6), C(6),

There are three paths that could solve "this problem" with the same cost.

- ACG = 7
- ADFG = 7
- ABG = 7