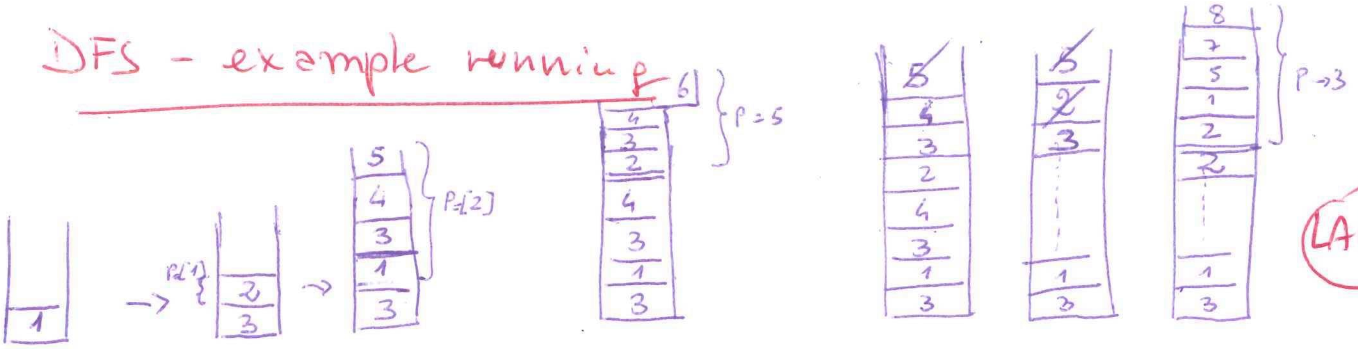
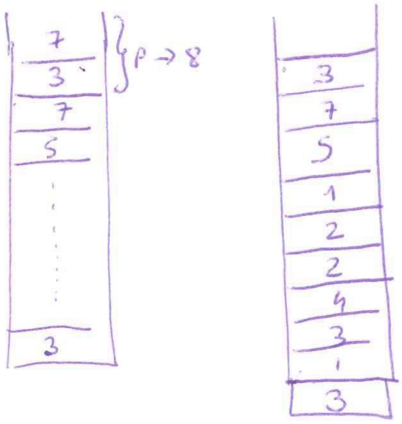
1) DFS - example running



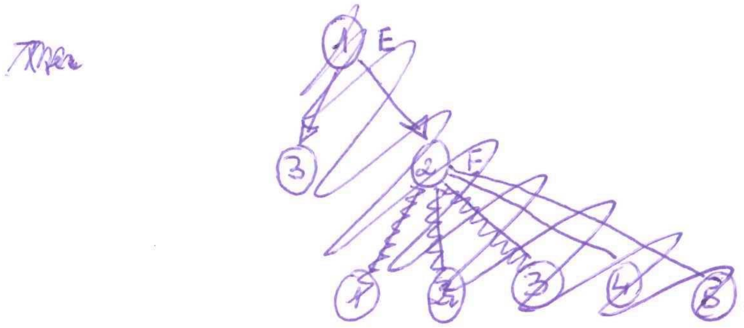
LAB1

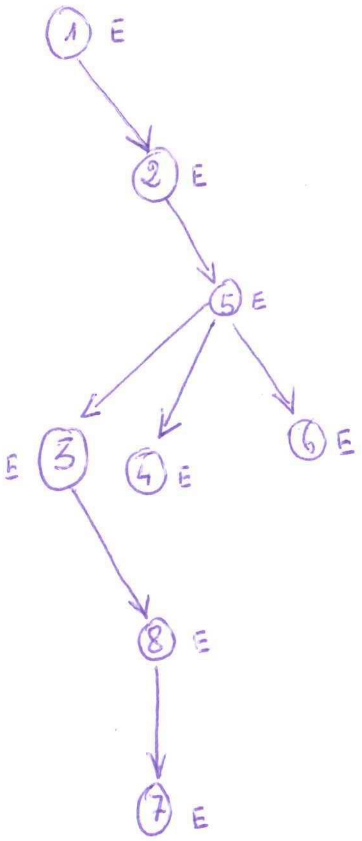


DFS Algorithm: Versione iterativa

input: see nex page

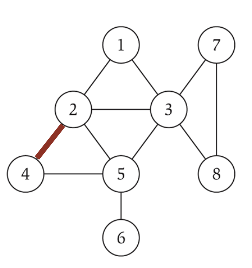
Output Spanning Tree

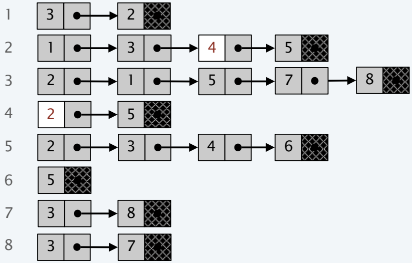




LAB 1

Graph Representation: Adjacency List





1 3 → 2

2 1 → 3 → 4 → 5

3 2 → 1 → 5 → 7 → 8

4 2 → 5

5 2 → 3 → 4 → 6

6 5

7 3 → 8

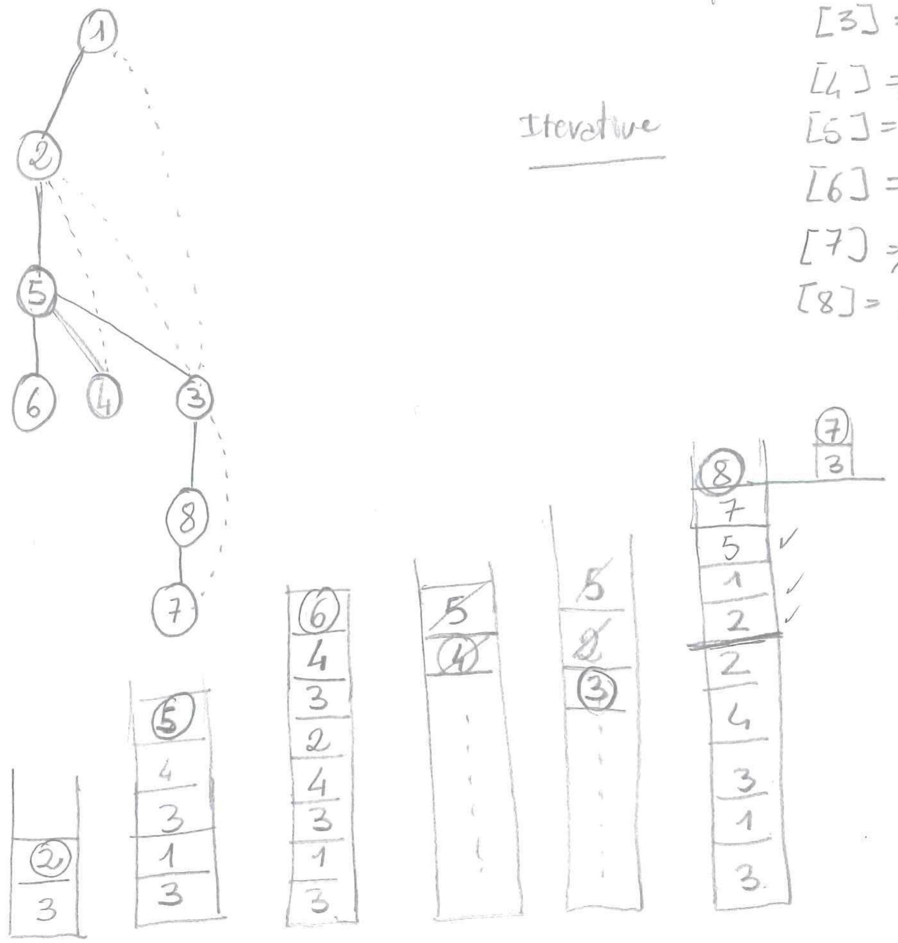
8 3 → 7

DFS

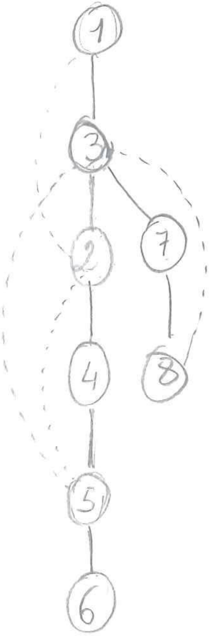
(Example from the slides-3)

Iterative

parent[2] = ~~1~~ 5
[3] = ~~2~~ 5
[4] = ~~2~~ 5
[5] = 2
[6] = 5
[7] = ~~3~~ 8
[8] = 3



Recursive



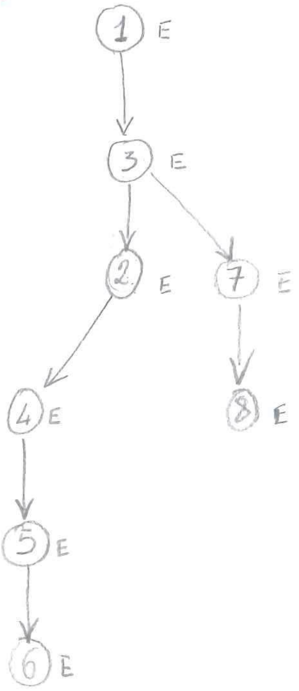
parent[3] = 1
" [2] = 3
[4] = 2
[5] = 4
[6] = 5
[7] = 3
[8] = 7

DFS (G,1)
  DFS (G,3)
    DFS~~(G,2)~~
      DFS~~(G,4)~~
        DFS~~(G,5)~~
          DFS~~(G,6)~~
  DFS (G,7)
  ~~DF~~ DFS (8)

LAB 1

2) • DFS - versione ricorsiva

Spanning tree



input:

1 ~~##~~: ~~3~~ → 2

2 : ~~1~~, ~~3~~, ~~4~~, ~~5~~

3 : ~~2~~, ~~1~~, ~~5~~, ~~7~~, 8

4 : ~~2~~, ~~5~~

5 : ~~2~~, ~~3~~, ~~4~~, ~~6~~

6 : ~~5~~

7 : ~~3~~, ~~8~~

~~Versione Ricorsiva~~

• Cycle Detection

Give an Algorithm to detect whether an undirected Graph has a cycle, Running Time should be $O(m+n)$

- Compute BFS
- ~~[illegible] second pass~~ Check whether by reaching an an already Discovered node, it has the same or next layer of the currently examined node.

```
Cycle-BFS(G)   initialize L[0]
  i=0;
  Discovered[V] = false, for each v∈V
  Layer[V] = 0 ,  for each v∈V
  For each v∈V
      If Discovered[v] = false then
          Layer[v] = i
          Add v to L[i];
          While L[i] ≠ 0
              Initialize L[i+1]
              For each u ∈ L[i]
                  For each (u,v) ∈ G
                      If Discovered[V] = F then
                          Disovered[V] = T
                          Layer[V] = i+1
                          Add v to L[i+1]
                      Else if(layer[v] = i OR
                              Layer[v] = i+1)
                          Return True
              i = i+1;
  return False;
```

• BFS & DFS

Show that if Spanning Tree - BS~~S~~ = Spanning Tree - DFS = T, then G = T.