



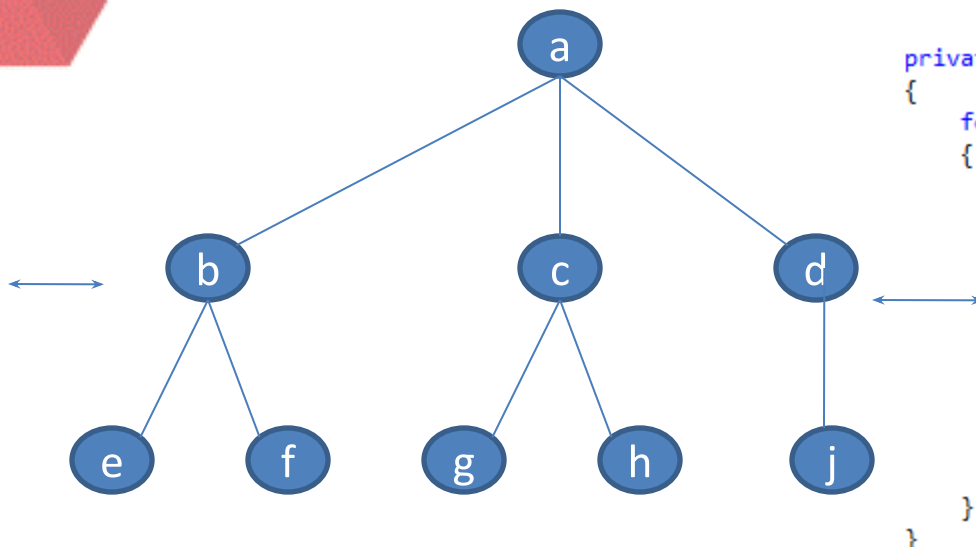
# PRETRAGE

Vežbe 2.



# Podsetnik sa predhodnih vežbi:

veza:a,b  
veza:a,c  
veza:a,d  
veza:b,e  
veza:b,f  
veza:c,g  
veza:c,h  
veza:d,j



```
private void formGraph(string[] lines)
{
    foreach (string line in lines)
    {
        string[] splitted = line.Split(':');
        string[] nodes = splitted[1].Split(',');
        Node node = new Node(nodes[0]);

        if (!graph.ContainsKey(node))
        {
            graph.Add(node, new List<Node>());
        }
        List<Node> nodeList = graph[node];
        nodeList.Add(new Node(nodes[1]));
    }
}
```

**Formiranje grafa iz Graph.txt (Vežbe 1)**



# Formiranje grafa

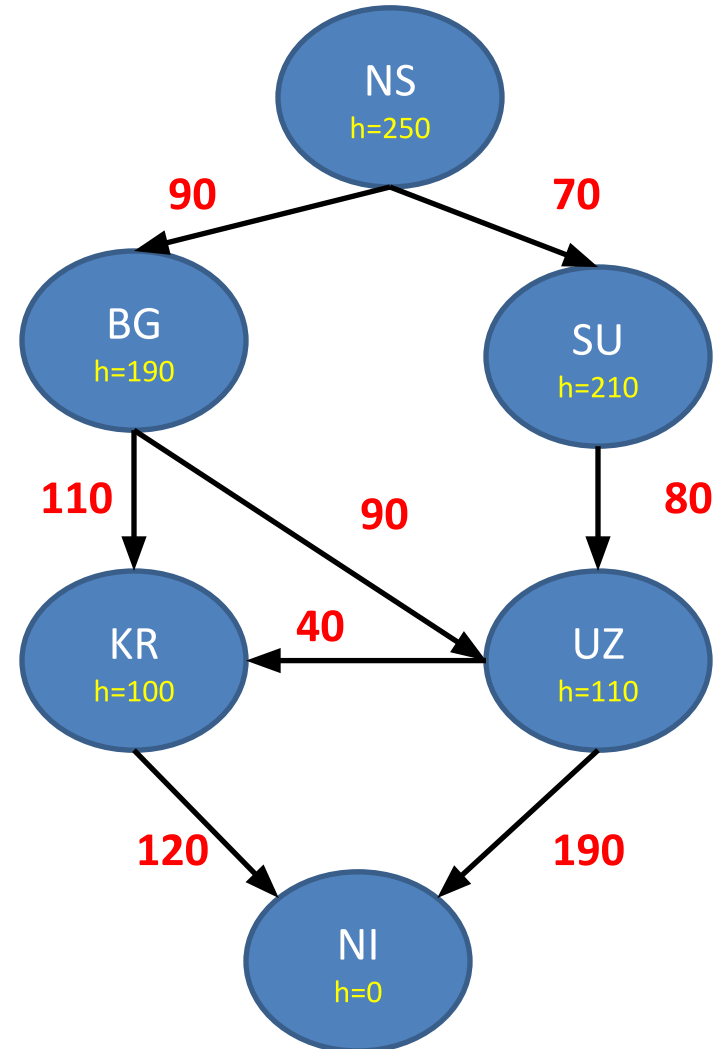
## Graph.txt

put:NS,BG,90  
put:NS,SB,70  
put:SB,UZ,80  
put:BG,UZ,90  
put:BG,KR,110  
put:UZ,KR,40  
put:UZ,NI,190  
put:KR,NI,120



## Nodes.txt

NS:250  
BG:190  
SB:210  
UZ:110  
KR:100  
NI:0





# Izmene u kodu

```
class Node
{
    public string Name { get; set; }
    public double Heuristic { get; set; }
    public List<Link> Links { get; set; }
```

**Heuristika i lista veza ka ostalim čvorovima.**

```
class State
{
    public State Parent { get; set; }
    public Node Node { get; set; }
    public double Cost { get; set; }
    public int Level { get; set; }
```

**Cost = ukupna cena stanja.  
Level = nivo čvora u grafu.**

```
class Graph
{
    public Dictionary<string, Node> graph = null;

    public Graph(string[] linesNodes, string[] linesLinks)
    {
        graph = new Dictionary<string, Node>();
        formGraph(linesNodes, linesLinks);
    }
}
```