$dp[v][2]$ — wynik gdy
$[0]$ — nie bierzemy
$[1]$ — bierzemy

```
dfs(s, p)
    for (v : g[s])
        if(v != p)
            dfs(v, s)
            dp[s][0] += dp[v][1]
```

```
dp[s][0]=dp[u][1]
dp[s][1]+=min(dp[u][0],
                       dp[u][1])

dp[s][1]+=cost[s]

dp[1][0..1]

odf(s,p,state)
    if(state && dp[s][0]<
                      dp[s][1])
        for(v:g[s])
            if(v!=p)
                odf(v,s,0)

    else
        res[s]=1
        odf(v,s,1)
```

1^0

2^1 3^0

5^1 -1

4^2 6^2

$1 \rightarrow 5$ $2 - 1 = 1$

1^{10}

5^2 7^{10}

3^{33} 6^{66}

3 4^{44} 5^{35} 8^{88} 0^{10}

$10^{10\,10}$

time_in
time_out

$A \quad x \quad y$

$V \quad x$
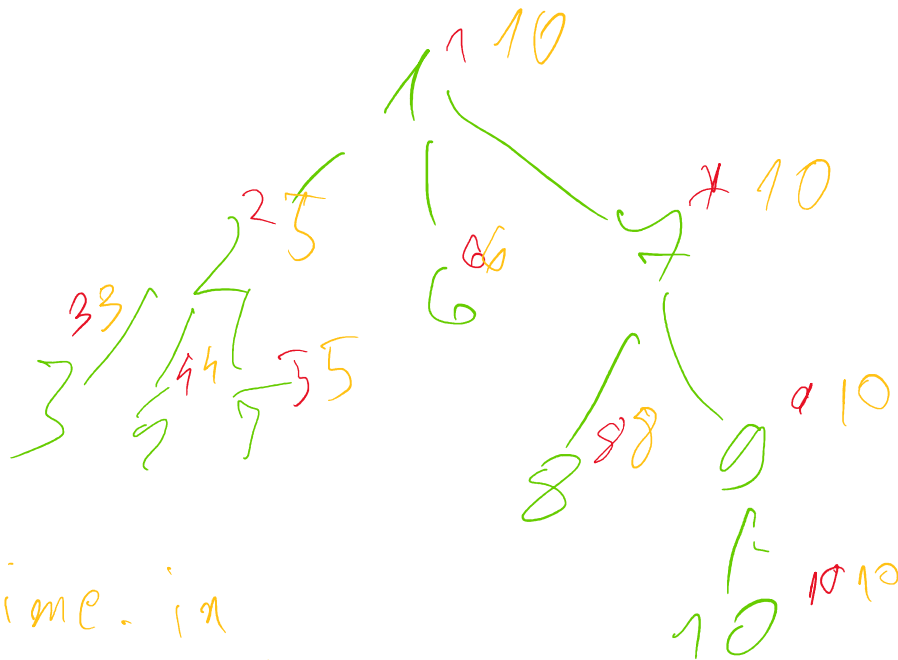
`if(depth[x] > depth[y]) swap`

`update (time_in[y],`
`time_out[y])`

`overh(x)`

$\lor \quad X$

$query(x)$