Characterize these roots?

**Thm 10**: $v$ is a root of a component iff when the DFS backtracks from $v$ no back-edges or cross-edges going out of its subtree (in the DFS-forest) to vertices still in the stack.

**Proof**: Assume we apply this rule then the following Invariants hold:

- The components we have already identified are correct
- If we backtracked from $v$ and $v$ is not a root then $v$ is in the same component as its lowest GRAY ancestor
How do we identify roots?

For each vertex $v$ compute $low(v) =$ the smallest $w \cdot d$ such that there is a cross edge or back edge to $w \in S$ out of the subtree of $v$. If there is no such edge then $low(v) = v \cdot d$.

If $low(v) = v \cdot d$ then $v$ is the root of a component.
How to compute low values?

Initialize $low(v) = v.d$

Update when you see a back edge or cross edge out of $v$

Update when you backtrack to $v$ from a child $w$ of $v$

$low(w)$ if smaller than $low(v)$ corresponds to a vertex that was discovered before $v$ which is on the stack and therefore a candidate for $low(v)$
Compute SCs using low values
Compute SCs using low values
הקשת זו לא גורמת עדכון כי היא שלחה לאעל הממסדית.
There is a mistake here, the node with discovery time 12 should change its low value to 7 and not to 6, see line 11 of the code on the last slide.
There is a mistake here, the node with discovery time 14 should change its low value to 12 and not to 6, see line 11 of the code on the last slide.
SCs using low values
## Tarjan’s SCC algorithm

**DFS(G)**

**SCC(G)**
1. For each vertex \( u \in G. V \)
2. \( u.\text{color} = \text{WHITE} \)
3. \( u.\pi = \text{NIL} \)
4. \( \text{time} = 0 \)
5. For each vertex \( u \in G. V \)
6. if \( u.\text{color} == \text{WHITE} \)
7. **SCC-Visit(G,u)**

**SCC-Visit(G,u)**

1. \( \text{time} = \text{time} + 1 \)
2. \( u.d = \text{low}(u) = \text{time} \)
3. push(u,S)
4. \( u.\text{color} = \text{GRAY} \)
5. For each vertex \( v \in G. \text{Adj}[u] \)
6. if \( v.\text{color} == \text{WHITE} \)
7. \( v.\pi = u \)
8. **SCC-Visit(G,v)**
9. \( \text{low}(u) = \min\{\text{low}(u), \text{low}(v)\} \)
10. else if \( v \in S \)
11. \( \text{low}(u) = \min\{v.d, \text{low}(u)\} \)
12. \( u.\text{color} = \text{BLACK} \)
13. \( \text{time} = \text{time} + 1 \)
14. \( u.f = \text{time} \)
15. if \( \text{low}(u) = u.d \) then perform pop(S)
16. until u is popped, the set popped is a SCC