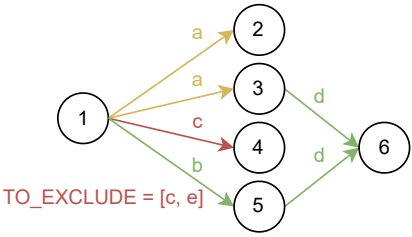
A.

1. Graph and exclusion list initialization



2. Go through every triplet, check conditions

1-a-2
1-a-3 } `1-a` occurs 2 times, choose randomly

1-c-4 } exclude, since `c` is banned

1-b-5 } include, this triplet is okay

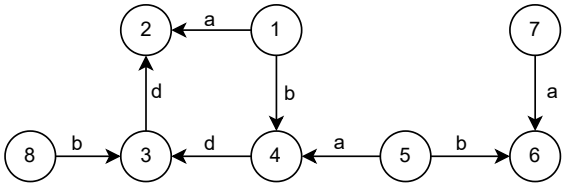
3-d-6
5-d-6 } these 2 are fine: <node>+<prop> are unique

3. Final list

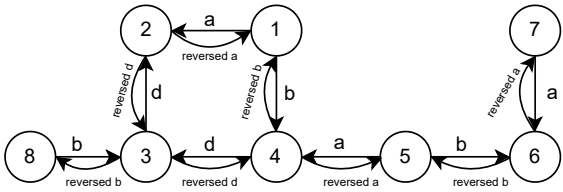
1-a-2
1-b-5
3-d-6
5-d-6

B.

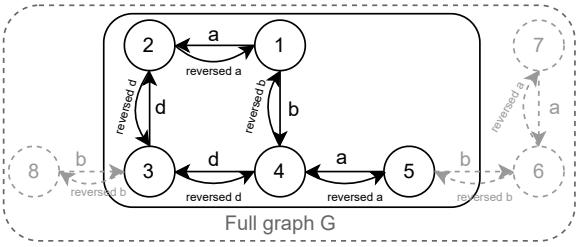
1. Initial graph: banned properties deleted



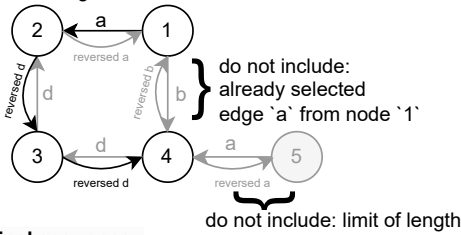
2. Extended graph: reversed properties added



3. Select subgraph, using BFS:
- from randomly selected node (here - `1`),
- depth (amount of hops) here = 2



4. Create sequence by traversing subgraph:
remember to check the uniqueness of <node+edge>
Stop when:
- # edges in sequence > predefined limit
- no valid neighbors



5. Final sequence:
`1` - `a` - `2 - `reversed d` - `3` - `reversed d` - `4`