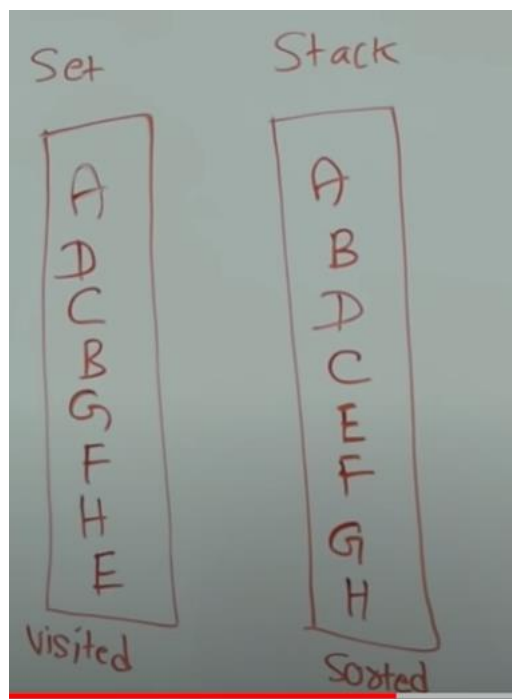


C should never before B



So one of the application of topological sort is in the build system suppose these were the packages of a build system saying that C has dependency on A and B and D has dependency on E and B packages. How would build system know which package it should build first? What it does is build a graph of packages and then applies topological sort on them like create one this

B D A C E F G

ordering so it creates packages. A and B builds first C has a dependency A and B then C can go next and E and D and F O go next et..

Let's see topological sort algorithm

Set has all visited vertices stack has all vertices in topological order.

1-We start from any node we start From E . E is is not on visited so we put E to visited set

2- Then we visit childs of E we come to H we want to visit childs of H but no childs so we are done with H then we write H to stack side.

3-Then we put F to visited set also it's child G G does not any children so we are done with G we put the to Stack(we finished to exploring that is why) then also F goes to Stack.

4-Then you can pick whatever value u want.We select B . we look at child of B and we see and C is not on visited set so we write also C has 1 child E but it is in the set already visited list so we don't do anything and C has no other child on this point we put C to stack. B also another child D . We put D also to visited set.D has a child of F but F is already in our visited list .we don't do anything.than we go to B at this point we explored B so B goes to Stack.

5- Then we go to A and we put A to visited list A has a one child C but we already in visited list.we done with A and write it to stack also

We sorted with topological sort algorithm

