

# Git: Rebase Your Expectations

## How to fixup your life

Josh Larson

February 21, 2019

# Photo Syncing

**Server**



**Laptop**

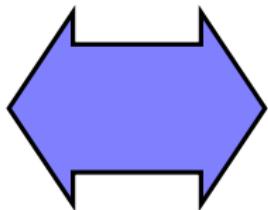


# Photo Syncing

Server



Laptop

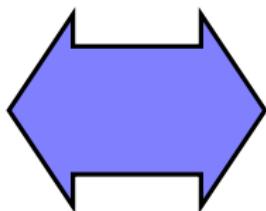


# Photo Syncing

Server



Laptop

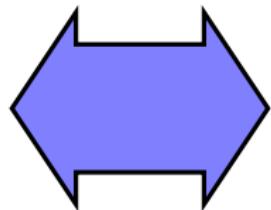


# Photo Syncing

Server



Laptop

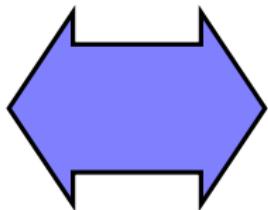


# Photo Syncing

Server



Laptop

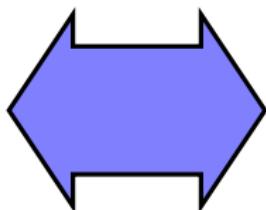


# Photo Syncing

Server



Laptop

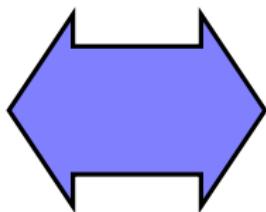


# Photo Syncing

Server



Laptop

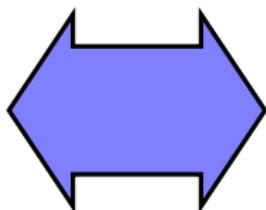


# Photo Syncing

Server



Laptop

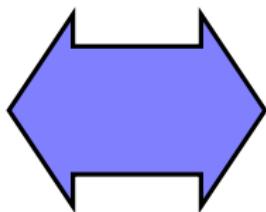


# Photo Syncing

Server



Laptop



## Photo Versions

- ▶ Changes

## Photo Versions

- ▶ Changes

## Photo Versions

- ▶ Changes
  - ▶ Add merchmart and shell

# Photo Versions

- ▶ Changes
  - ▶ Add merchmart and shell



# Photo Versions

- ▶ Changes
  - ▶ Add merchmart and shell
  - ▶ Add skyline



# Photo Versions

- ▶ Changes
  - ▶ Add merchmart and shell
  - ▶ Add skyline



# Photo Versions

- ▶ Changes
  - ▶ Add merchmart and shell
  - ▶ Add skyline
  - ▶ Remove merchmart



# Photo Versions

- ▶ Changes
  - ▶ Add merchmart and shell
  - ▶ Add skyline
  - ▶ Remove merchmart



# Photo Versions

**Server**



**Laptop**



# Photo Versions

## Server

- ▶ Add 2 images



## Laptop

- ▶ Add 2 images



# Photo Versions

## Server

- ▶ Add 2 images
- ▶ Add merchmart



## Laptop

- ▶ Add 2 images



# Photo Versions

## Server

- ▶ Add 2 images
- ▶ Add merchmart



## Laptop

- ▶ Add 2 images
- ▶ Add merchmart



# Photo Versions

## Server

- ▶ Add 2 images
- ▶ Add merchmart



## Laptop

- ▶ Add 2 images
- ▶ Add merchmart



# Photo Versions

**Server**



**Laptop**



# Photo Versions

## Server

- ▶ Add 3 images



## Laptop

- ▶ Add 3 images



# Photo Versions

## Server

- ▶ Add 3 images



## Laptop

- ▶ Add 3 images
- ▶ Remove merchmart



# Photo Versions

## Server

- ▶ Add 3 images
- ▶ Remove merchmart



## Laptop

- ▶ Add 3 images
- ▶ Remove merchmart



# Photo Versions

## Server

- ▶ Add 3 images
- ▶ Remove merchmart



## Laptop

- ▶ Add 3 images
- ▶ Remove merchmart



# Commits

# What is a Commit?

## What is a Commit?

- ▶ “Apply **these changes** to **this universe** (and some **other stuff**)”

## What is a Commit?

- ▶ “Apply **these changes** to **this universe** (and some **other stuff**)”
  - ▶ “**these changes**”: The diff

## What is a Commit?

- ▶ “Apply **these changes** to **this universe** (and some **other stuff**)”
  - ▶ “**these changes**”: The diff
  - ▶ “**this universe**”: Parent commit

## What is a Commit?

- ▶ “Apply **these changes** to **this universe** (and some **other stuff**)”
  - ▶ “**these changes**”: The diff
  - ▶ “**this universe**”: Parent commit
  - ▶ “**other stuff**”: Metadata, like author, timestamp, and commit message

## What is a Commit?

- ▶ “Apply **these changes** to **this universe** (and some **other stuff**)”
  - ▶ “**these changes**”: The diff
  - ▶ “**this universe**”: Parent commit
  - ▶ “**other stuff**”: Metadata, like author, timestamp, and commit message
- ▶ a8808ba20c1d3d3f26f685d71a29de3e452d82711

## What is a Commit?

- ▶ “Apply **these changes** to **this universe** (and some **other stuff**)”
  - ▶ “**these changes**”: The diff
  - ▶ “**this universe**”: Parent commit
  - ▶ “**other stuff**”: Metadata, like author, timestamp, and commit message
- ▶ a8808ba20c1d3d3f26f685d71a29de3e452d82711
- ▶ a8808ba

History?

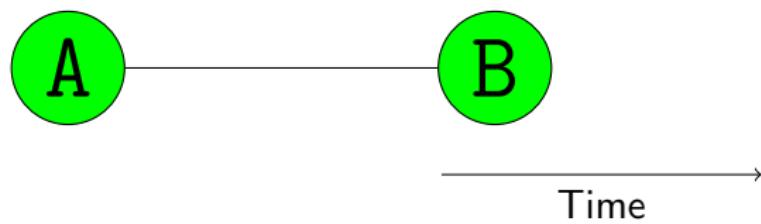
# Intent

# Ancestry

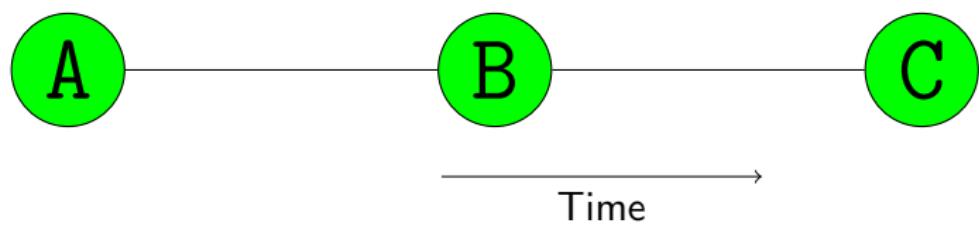
# Ancestry



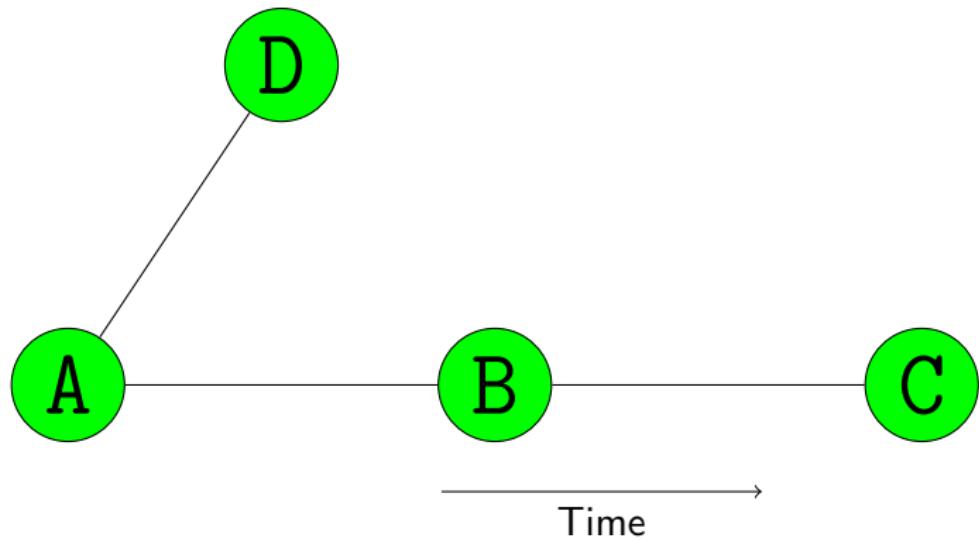
# Ancestry



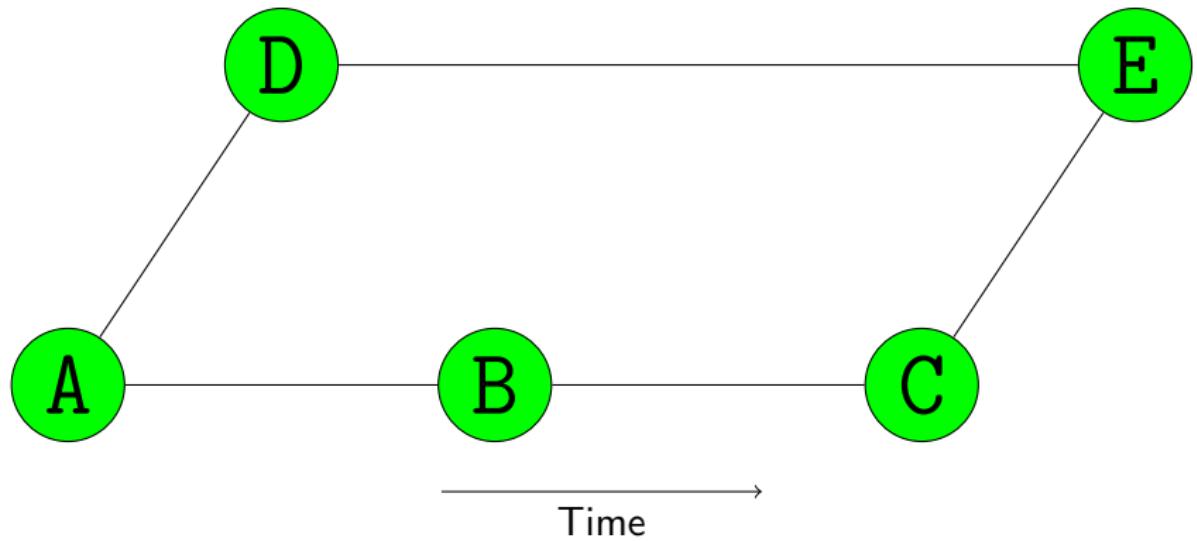
# Ancestry



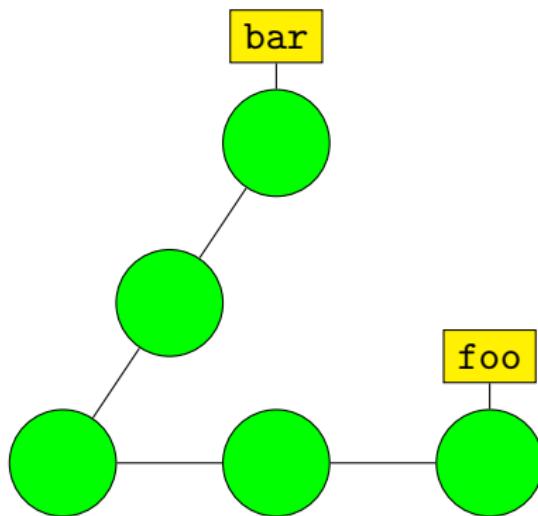
# Ancestry



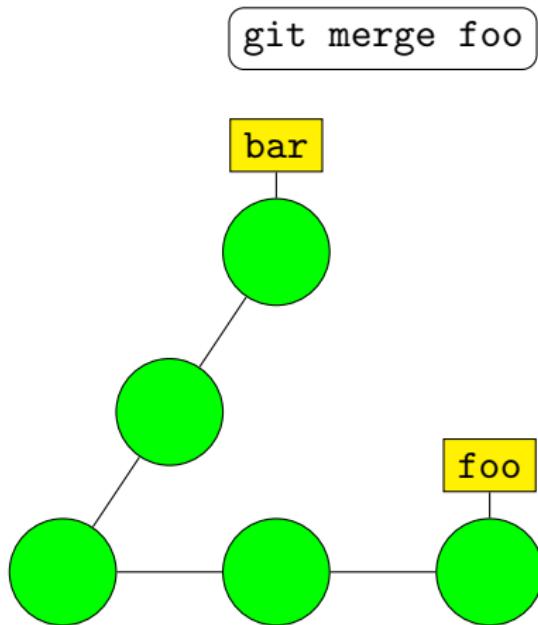
# Ancestry



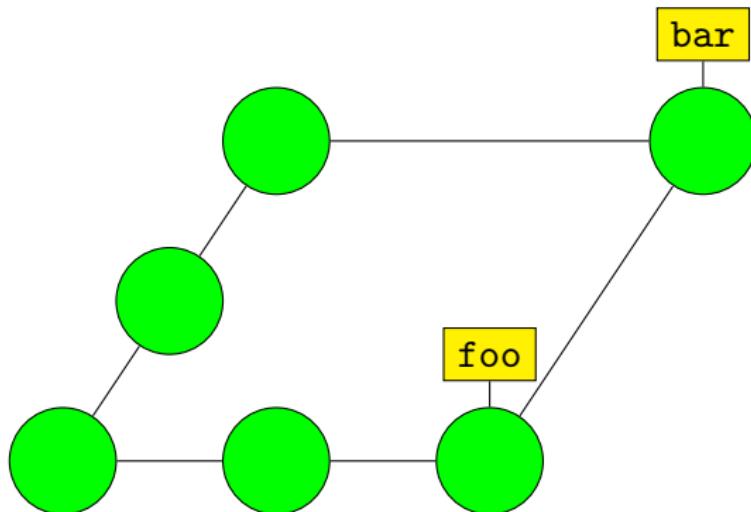
# Merging



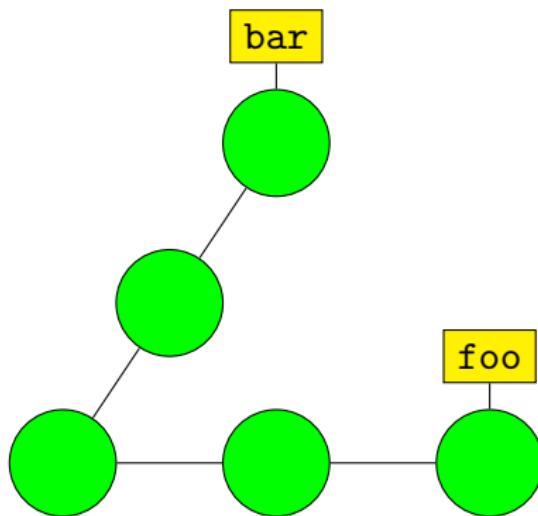
# Merging



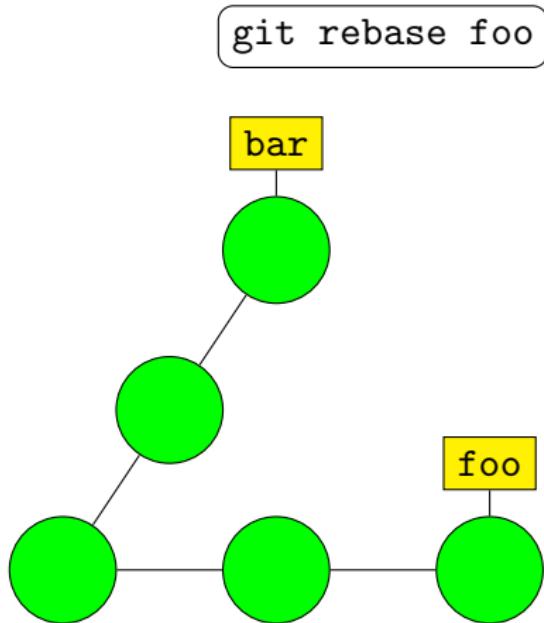
# Merging



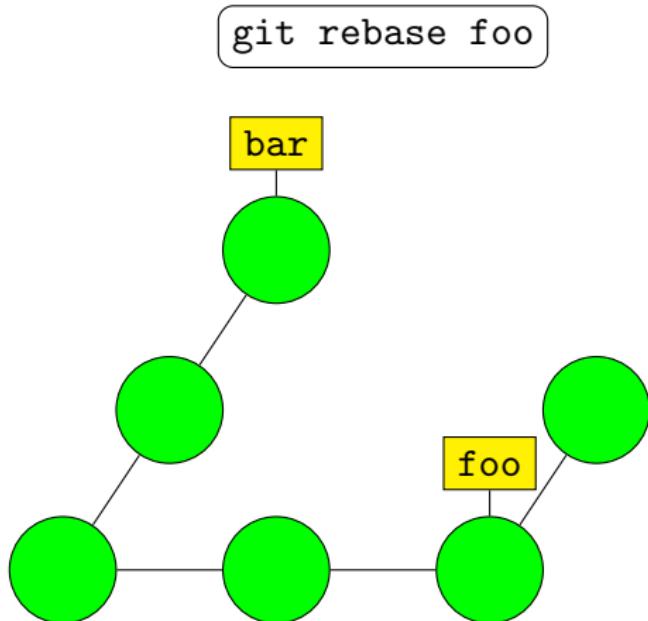
# Merging



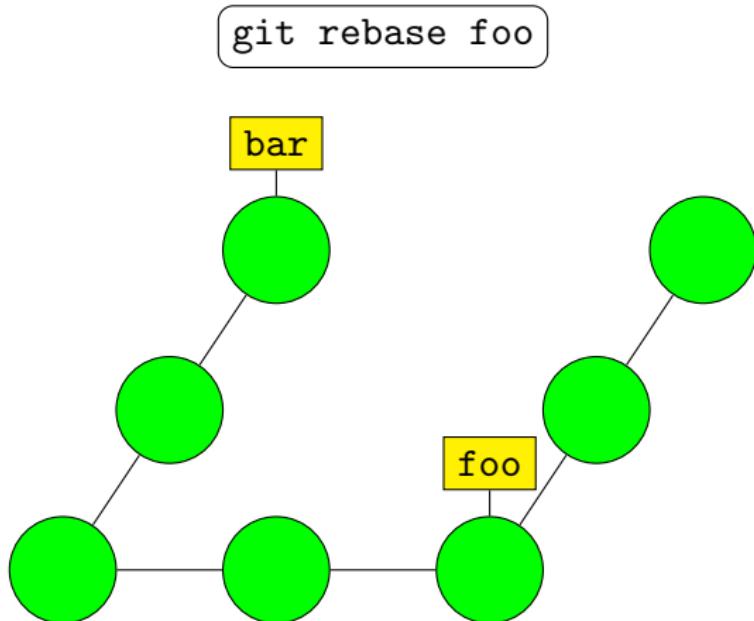
# Merging



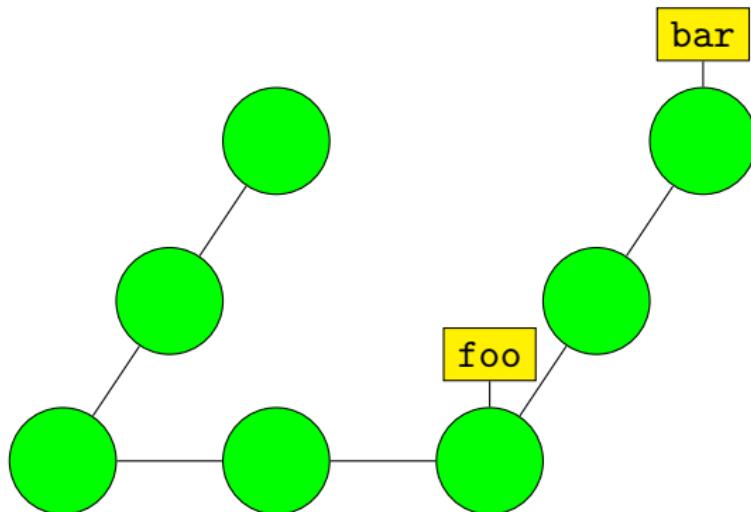
# Merging



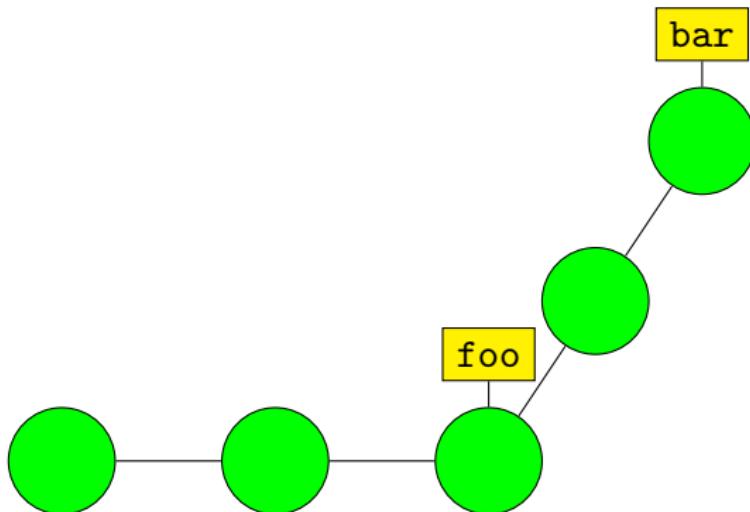
# Merging



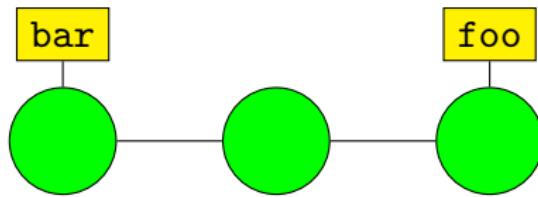
# Merging



# Merging

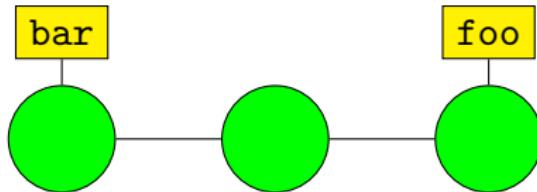


# Merging

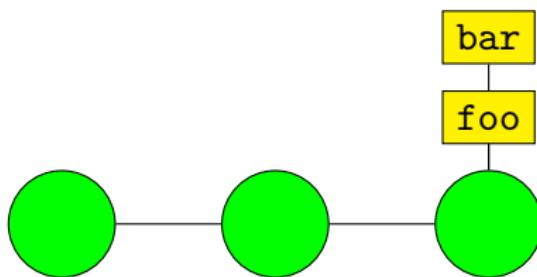


# Merging

git merge foo

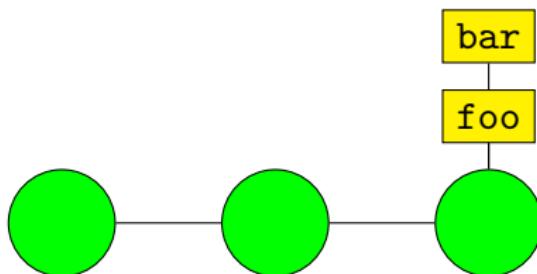


# Merging



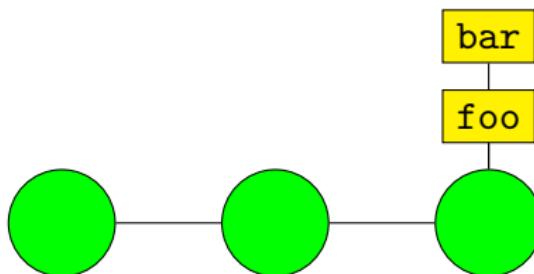
# Merging

```
git rebase foo
```

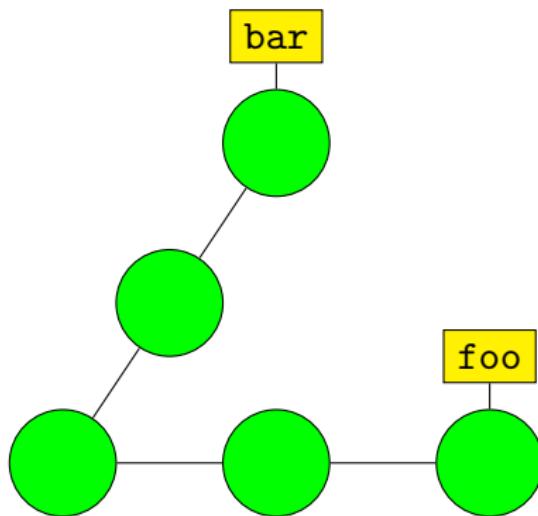


# Merging

```
git merge --ff-only foo
```

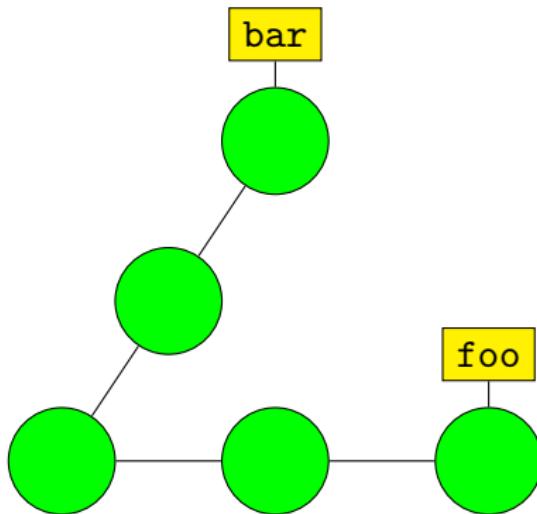


# Merging

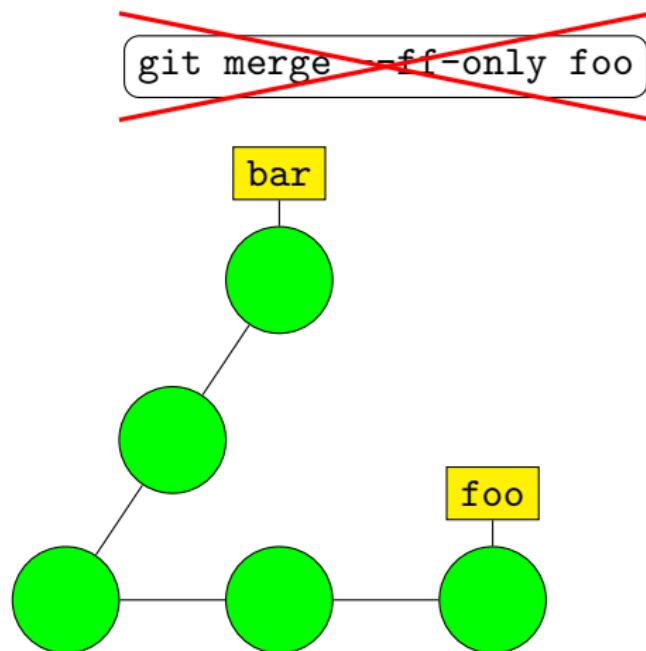


# Merging

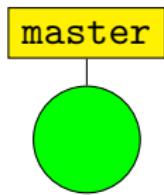
```
git merge --ff-only foo
```



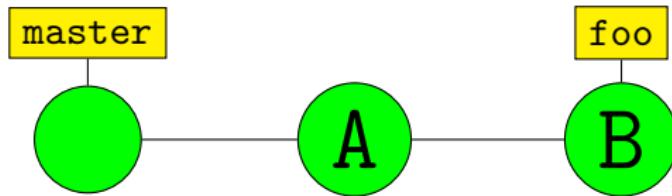
# Merging



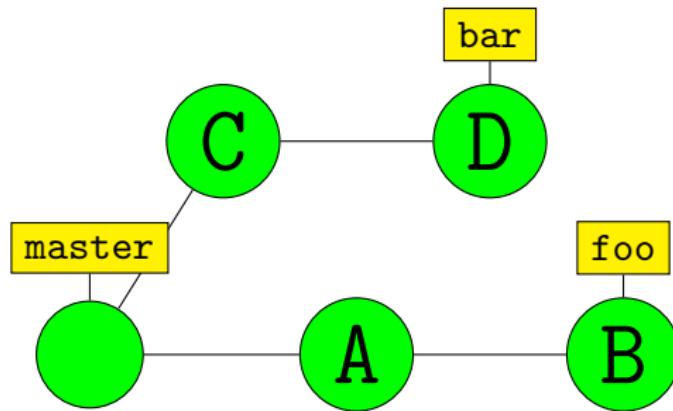
# Merge Strategy



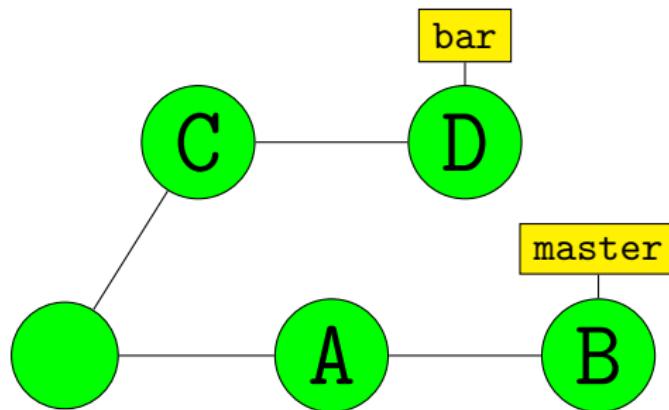
# Merge Strategy



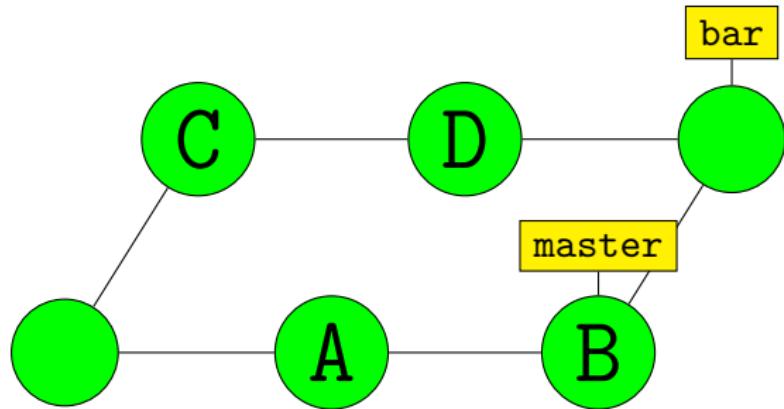
# Merge Strategy



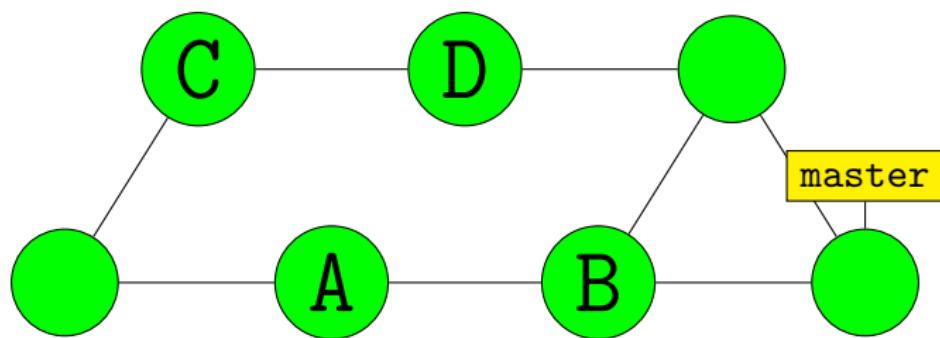
## Merge Strategy



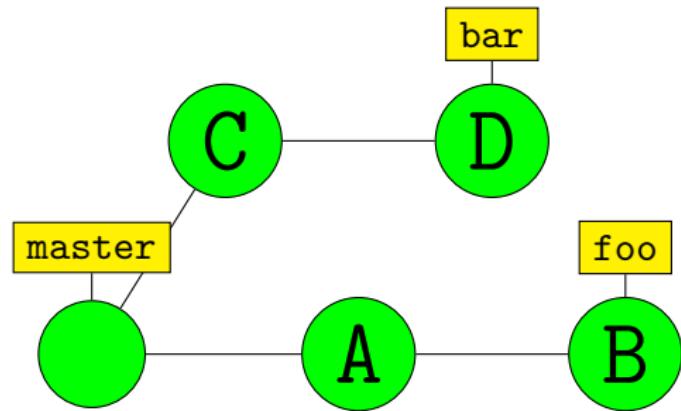
## Merge Strategy



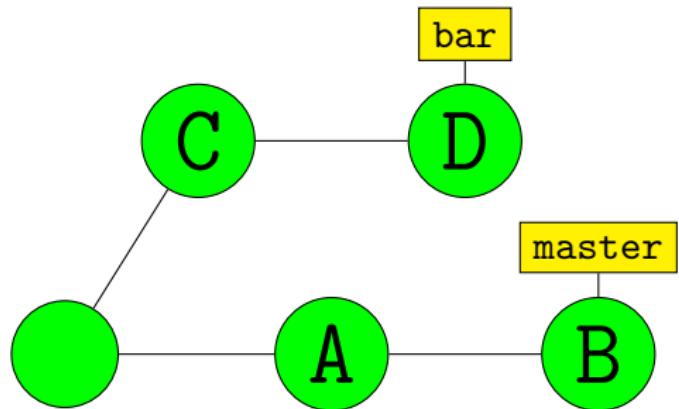
## Merge Strategy



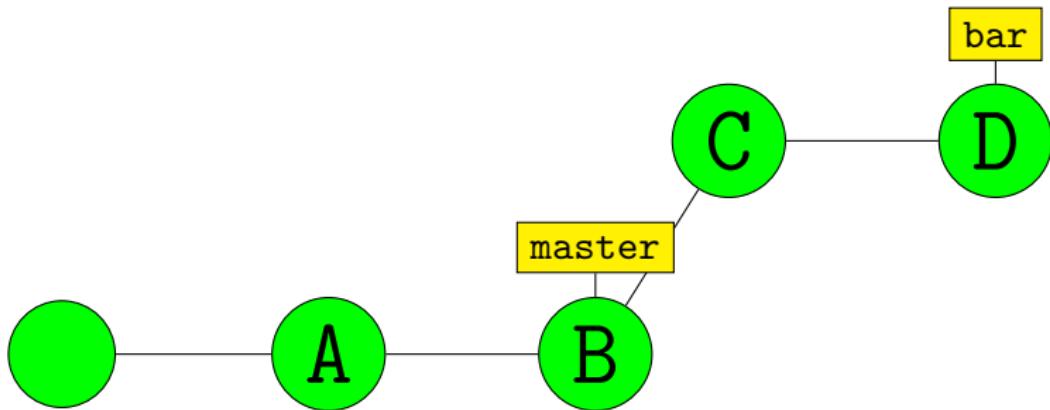
## Merge Strategy (Rebase)



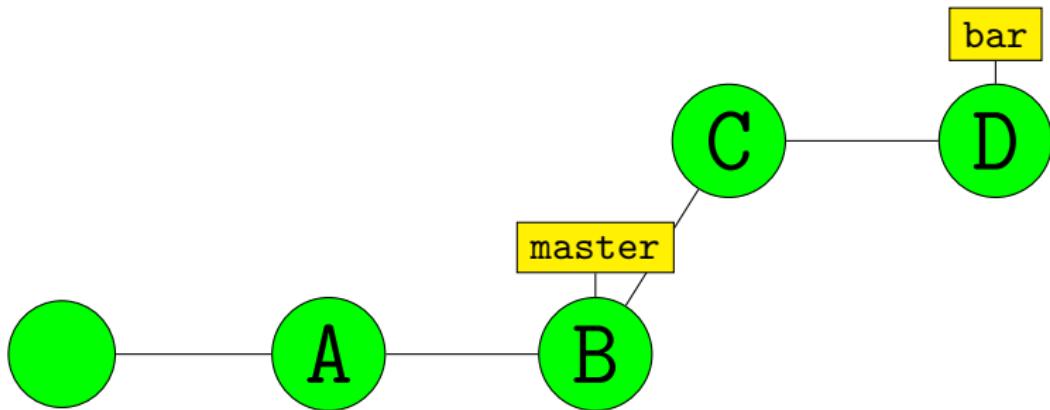
## Merge Strategy (Rebase)



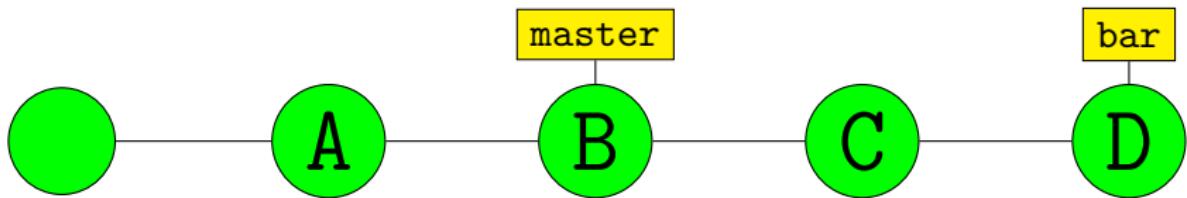
## Merge Strategy (Rebase)



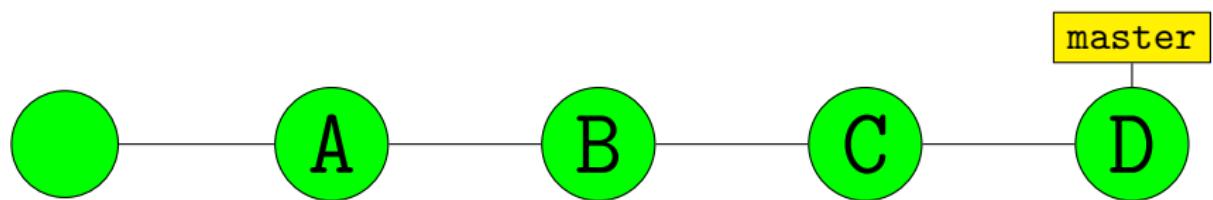
## Merge Strategy (Rebase)



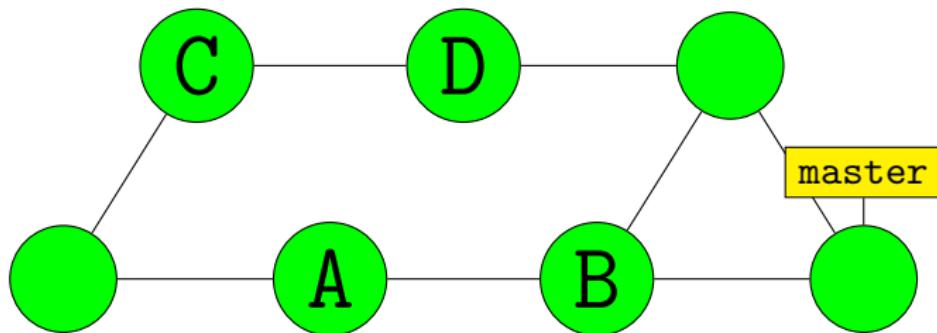
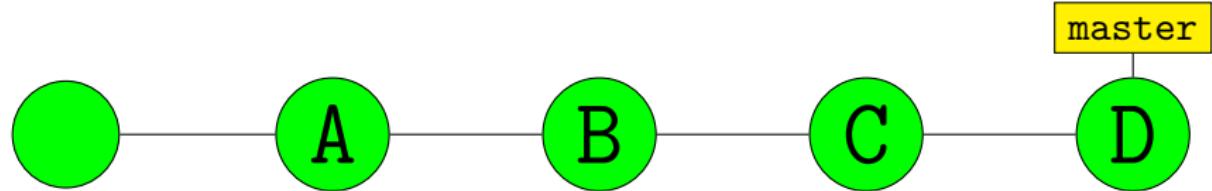
## Merge Strategy (Rebase)



## Merge Strategy (Rebase)



## Merge Strategies



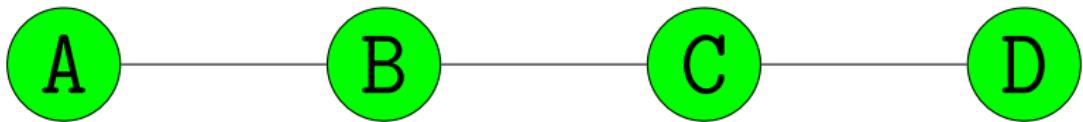
# Rewriting History



```
git rebase -i A
```



pick B  
pick C  
pick D



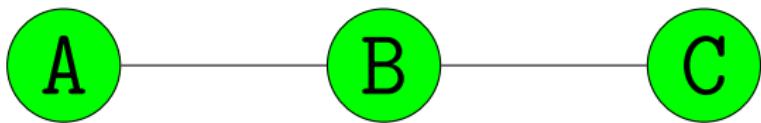
pick B  
 pick C  
 pick D

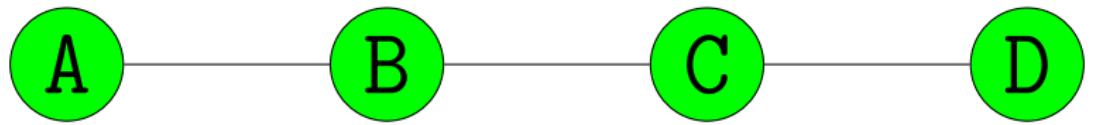
A



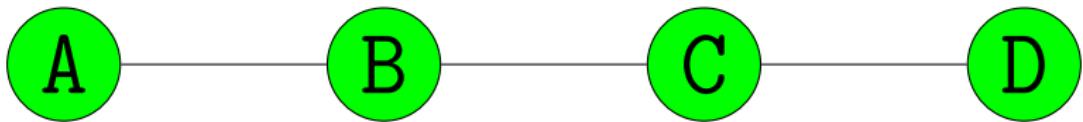
pick C  
pick D

pick D

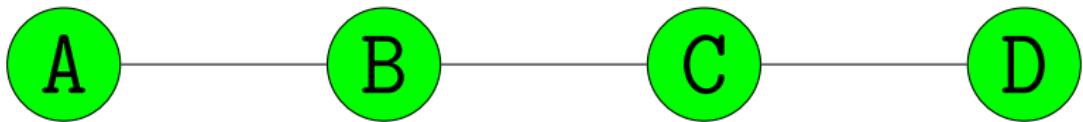




pick B  
pick C  
pick D



pick C  
pick B  
pick D



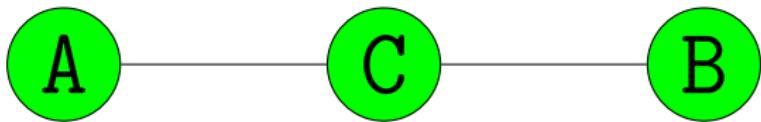
pick C  
 pick B  
 pick D

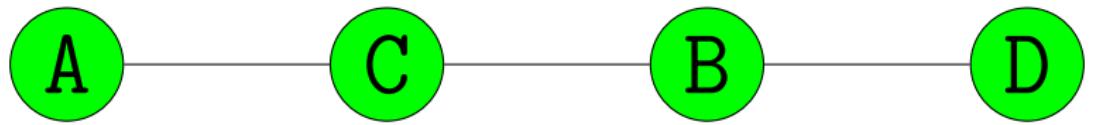
A



pick B  
pick D

pick D

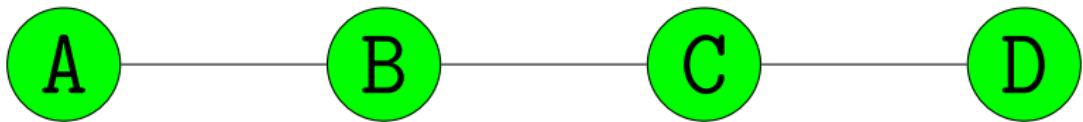




pick B  
pick C  
pick D



pick B  
pick D  
pick C



pick B  
fixup D  
pick C



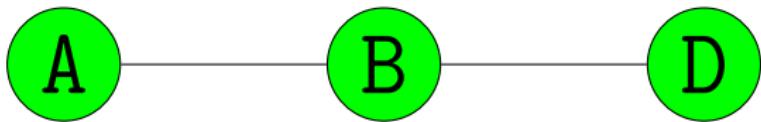
pick B  
fixup D  
pick C

A

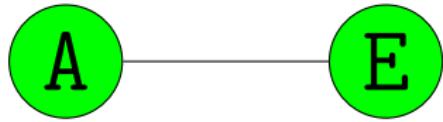
fixup D  
pick C

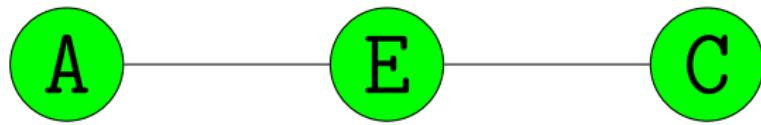


fixup  
pick C



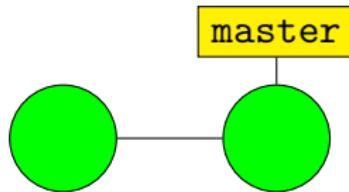
pick C



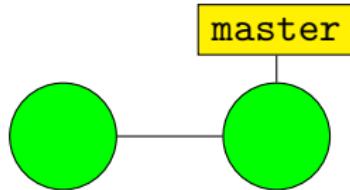


# Pitfalls

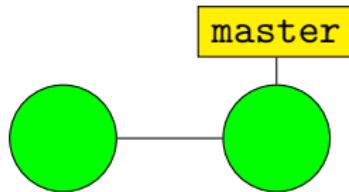
# Remotes



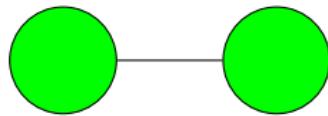
.....

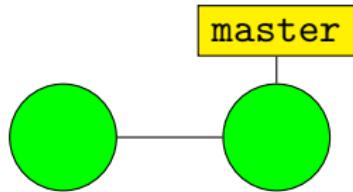


```
git clone path@to:remote/repo
```

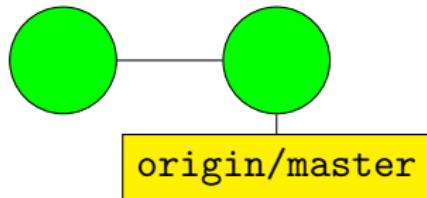


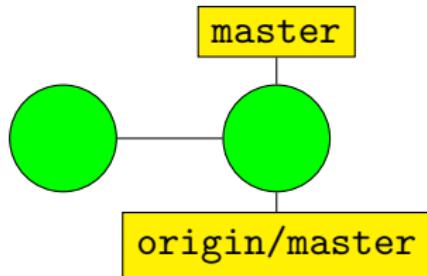
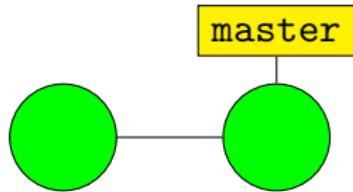
.....

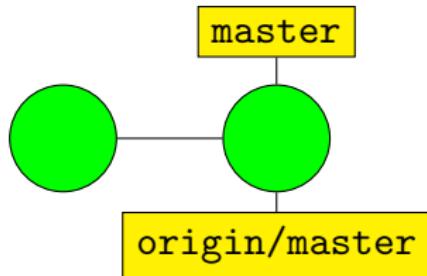
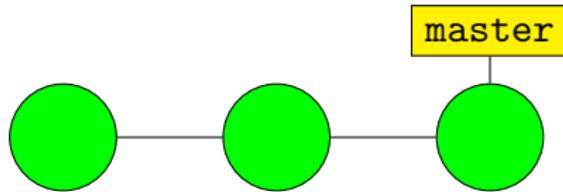


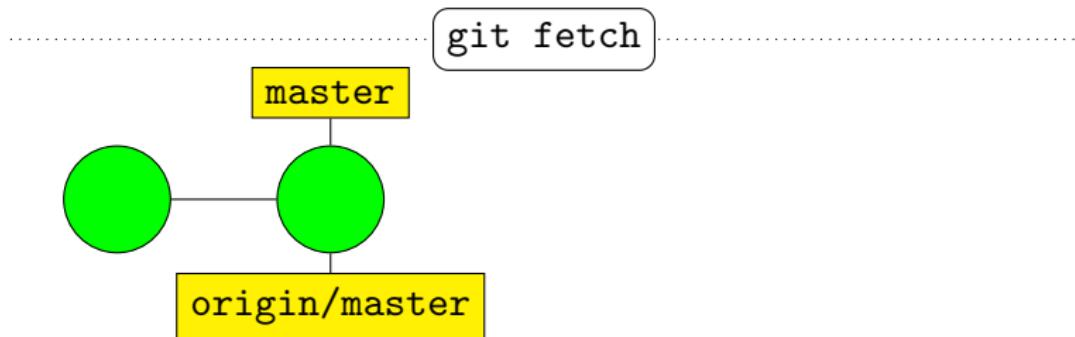
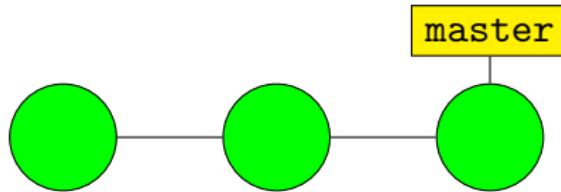


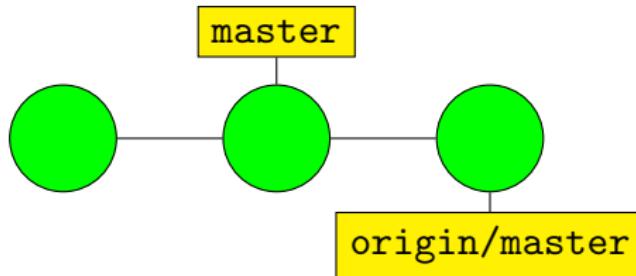
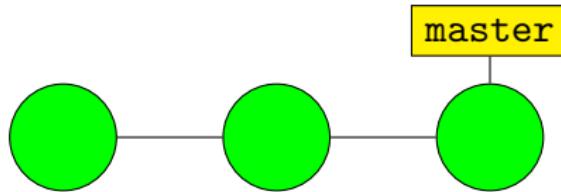
.....

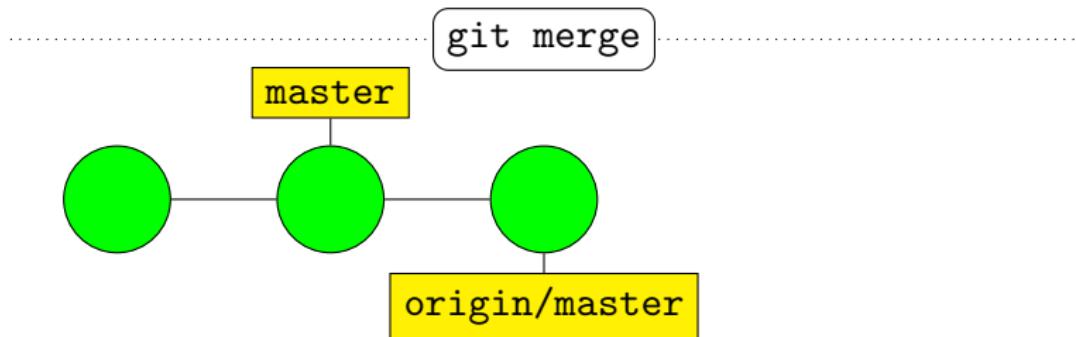
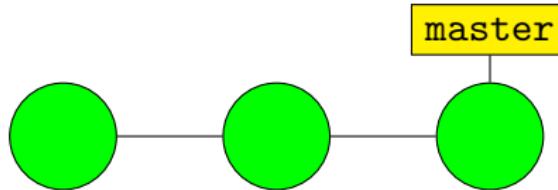


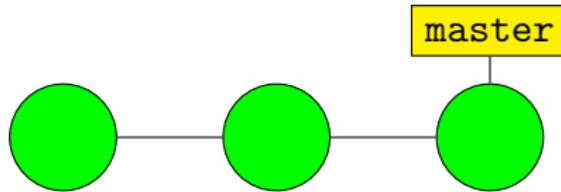




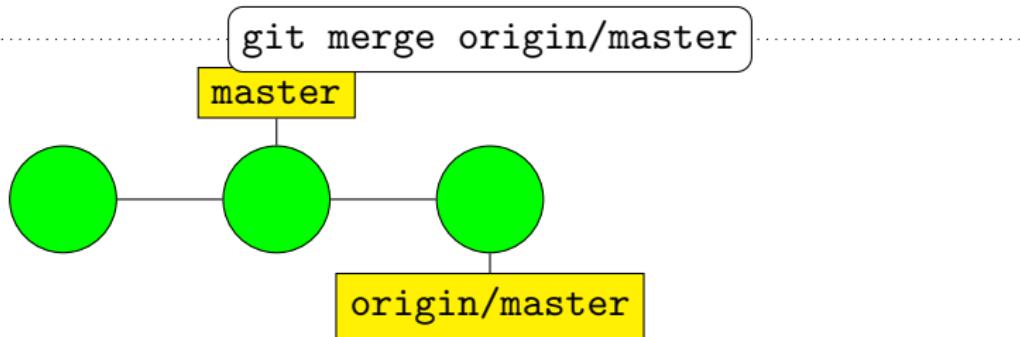








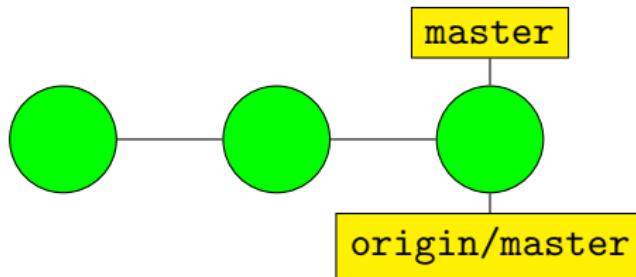
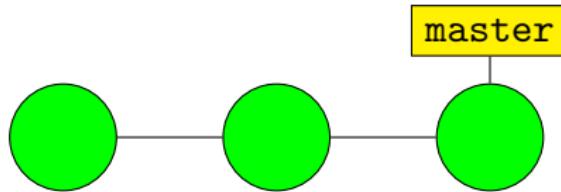
master



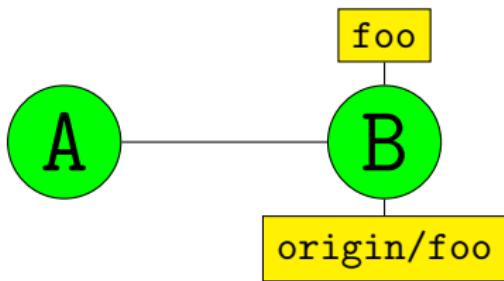
git merge origin/master

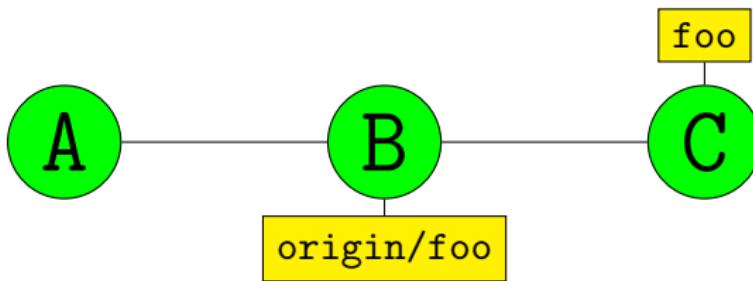
master

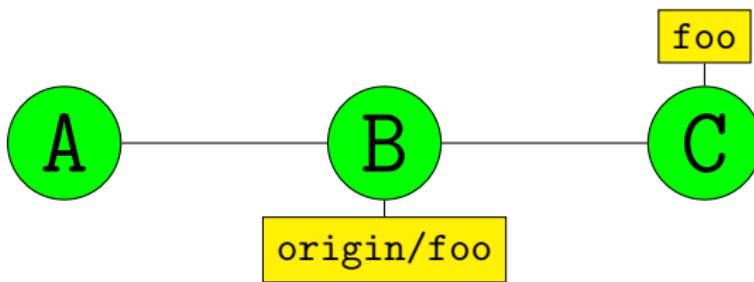
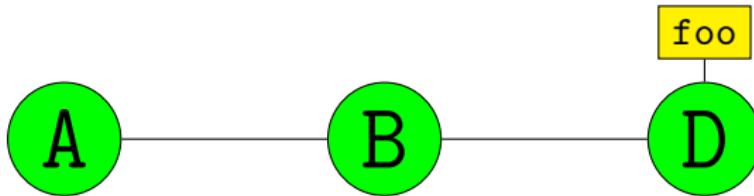
origin/master

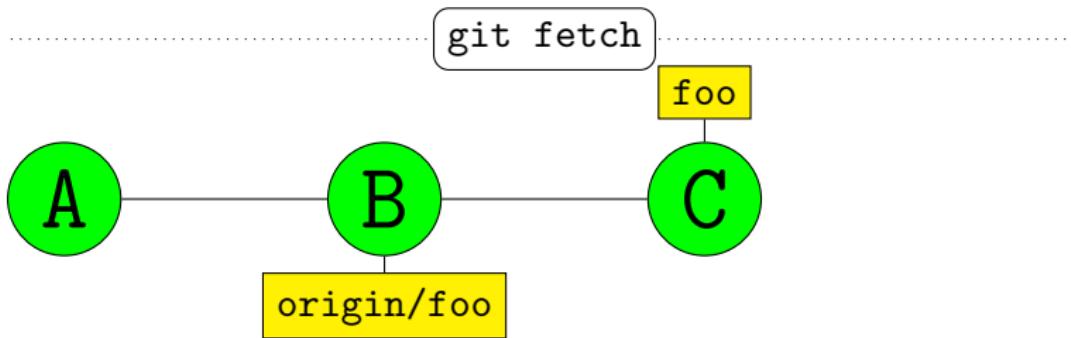
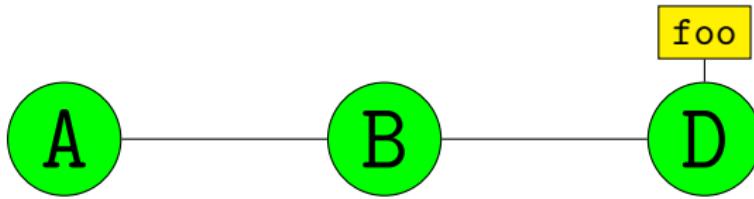


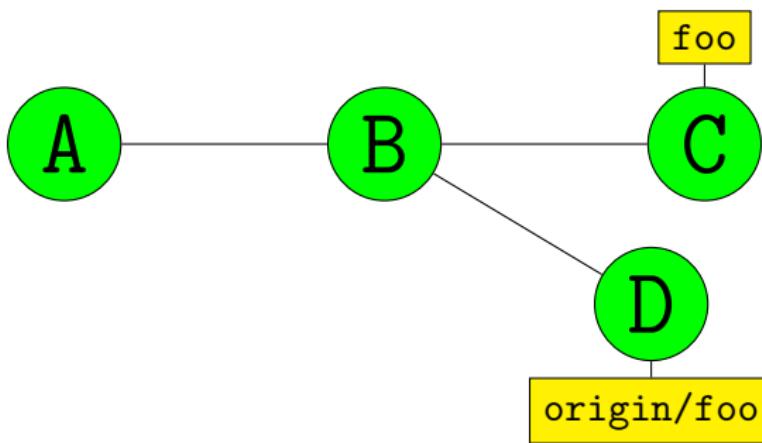
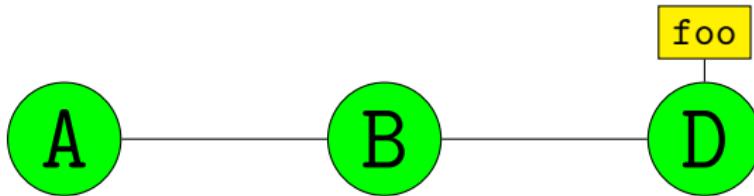
# Collaborating

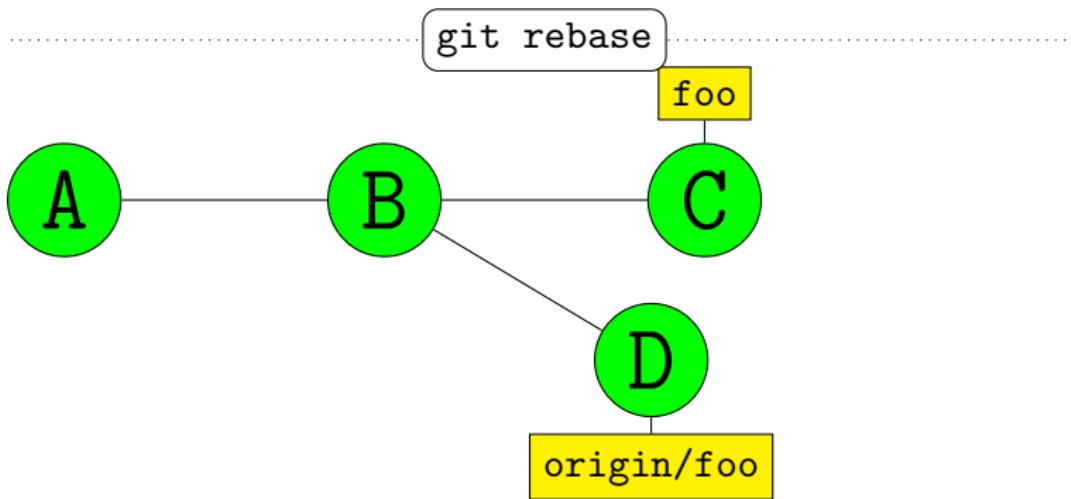
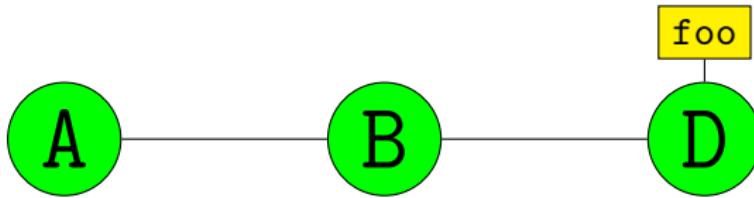


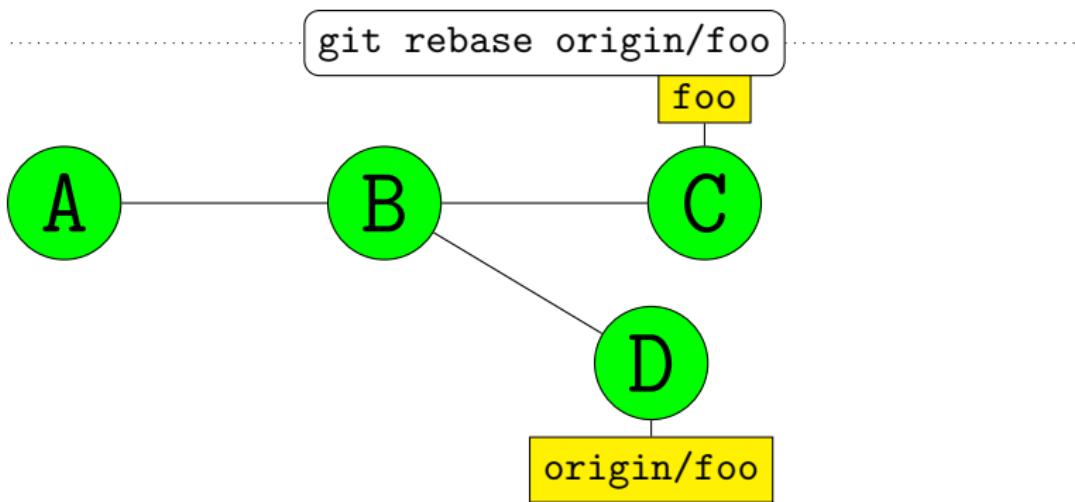
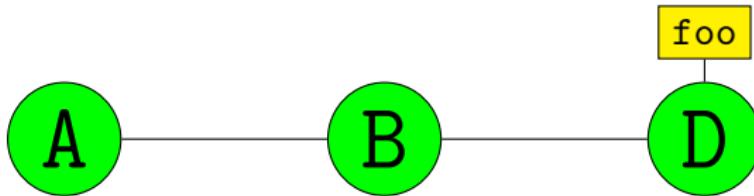


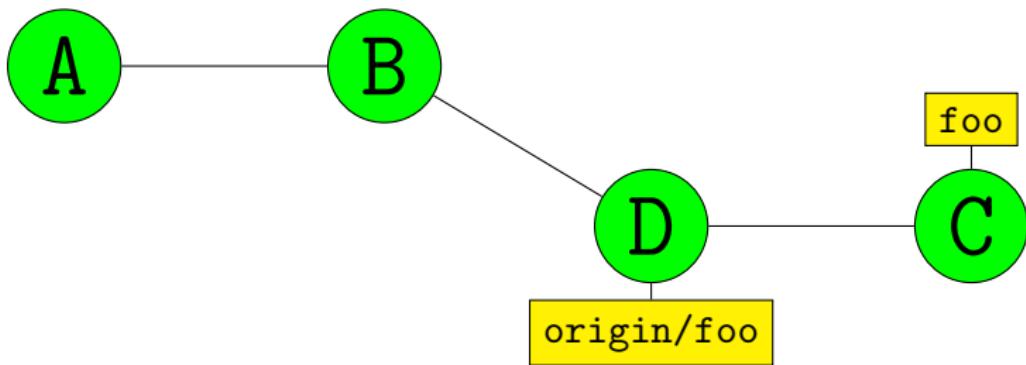
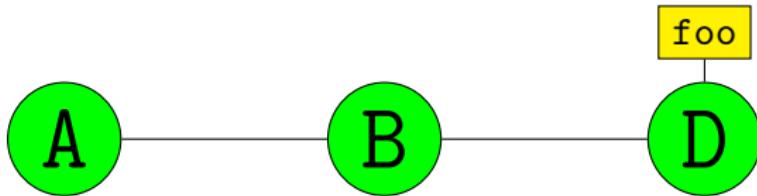


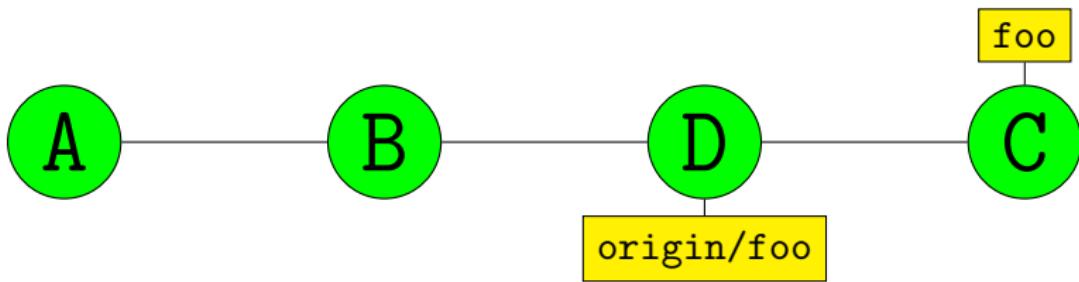
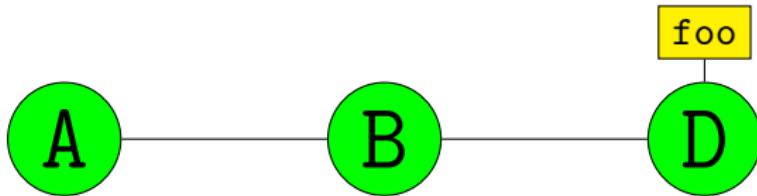


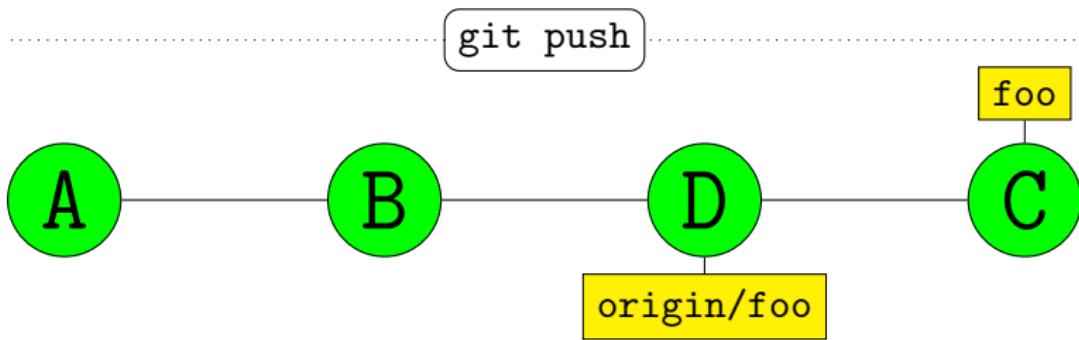
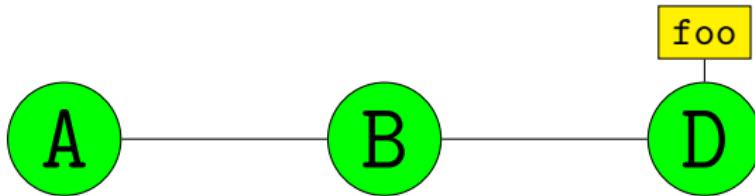


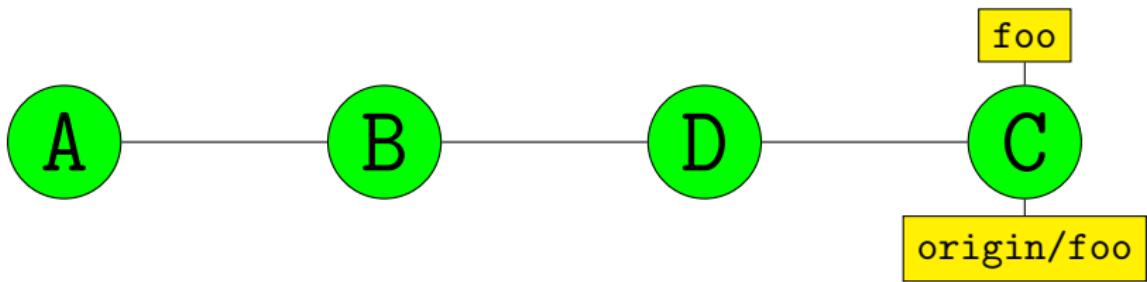


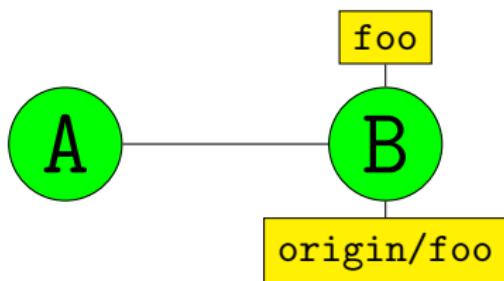


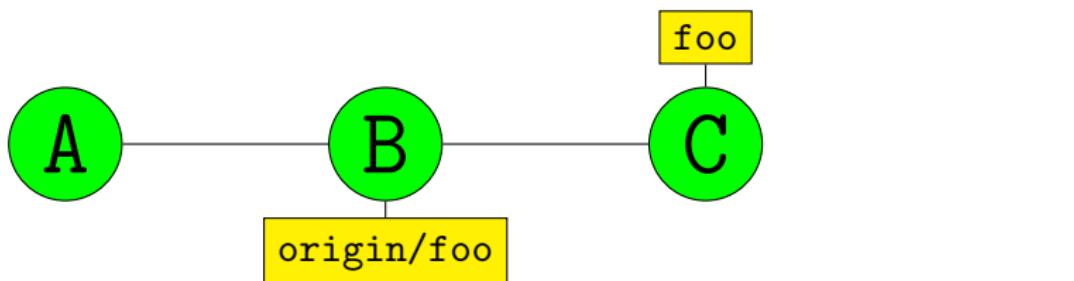


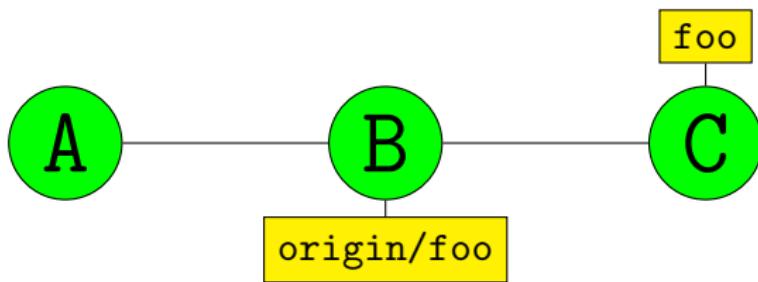
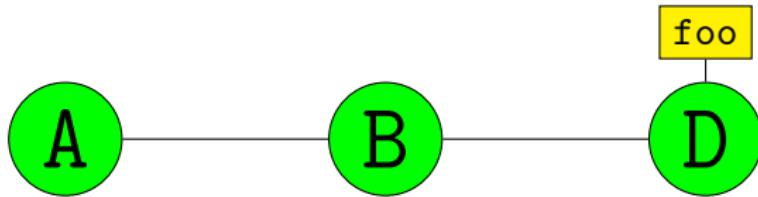


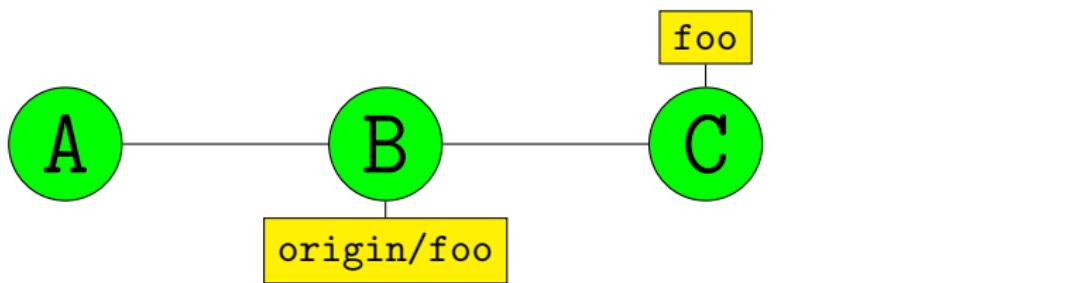


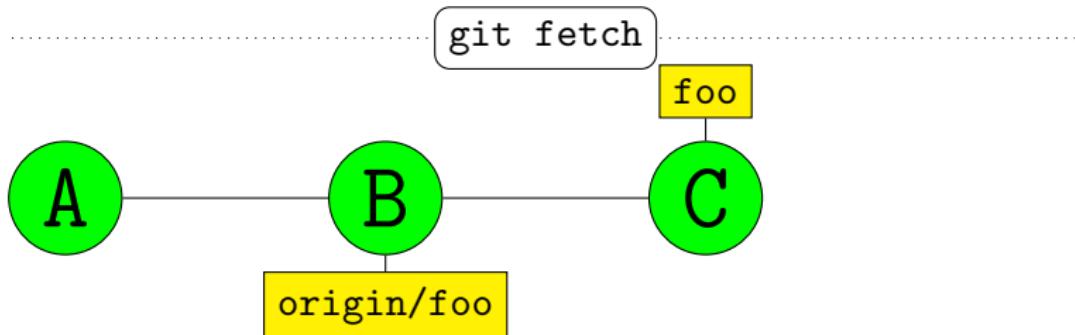


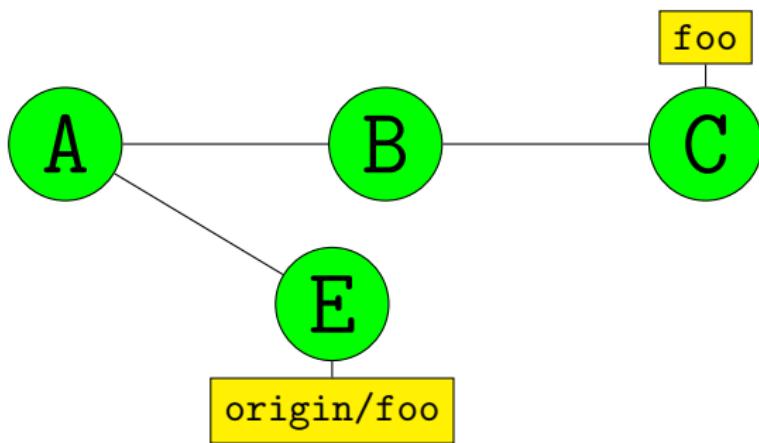


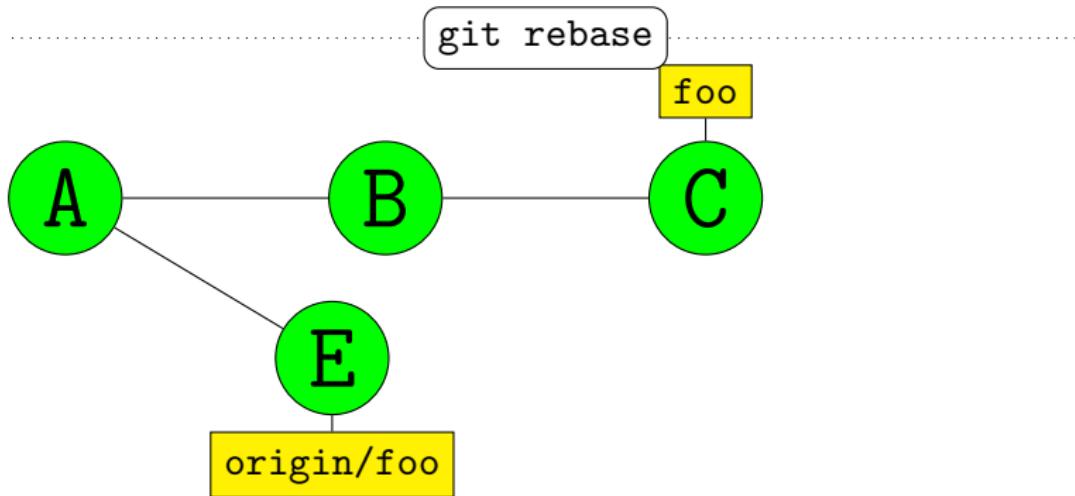


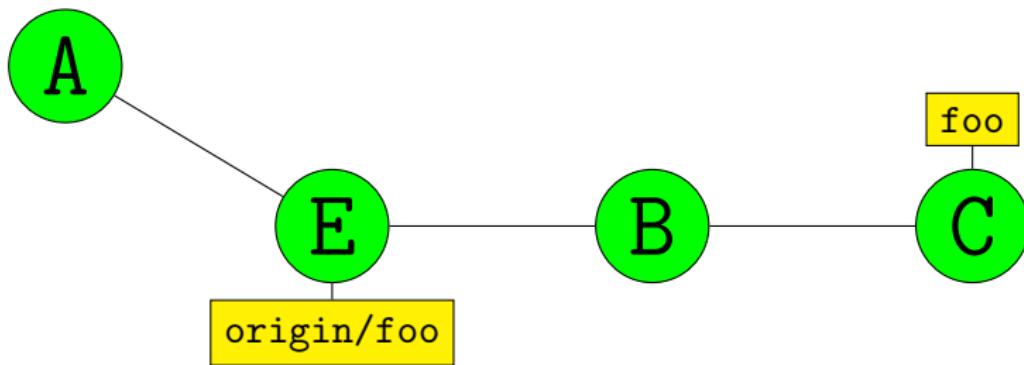


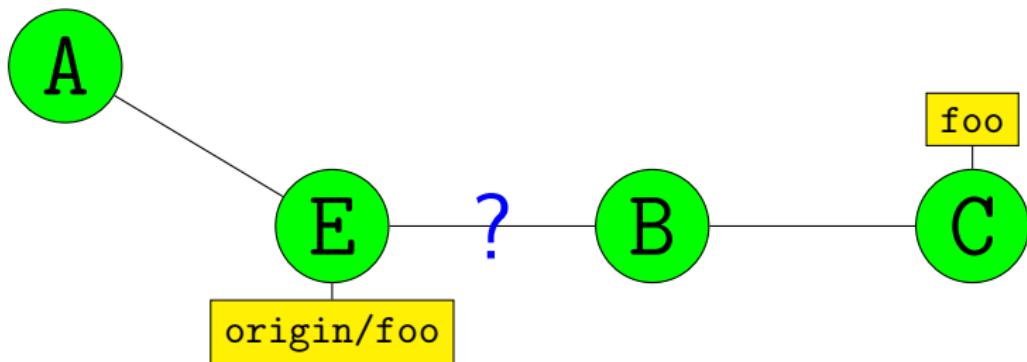








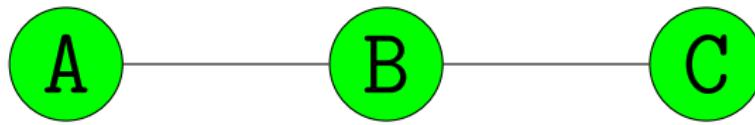




# Rewriting History?

Carefully

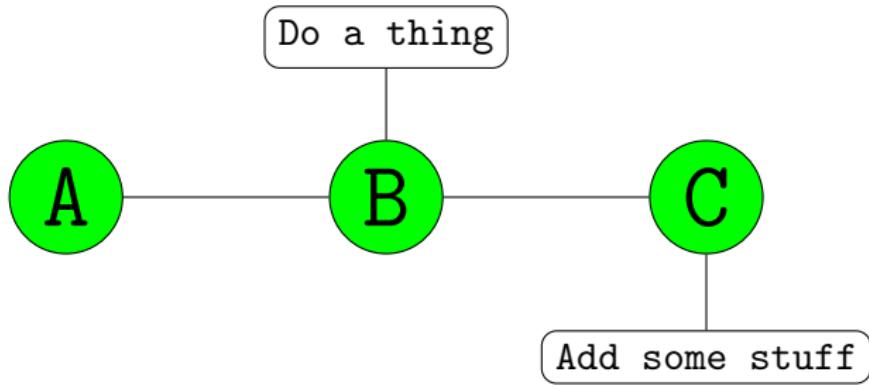
# Fixup Commits

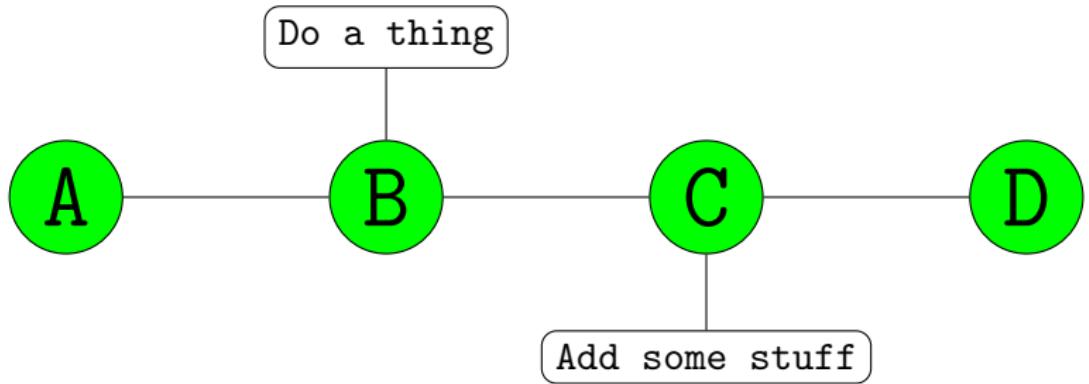


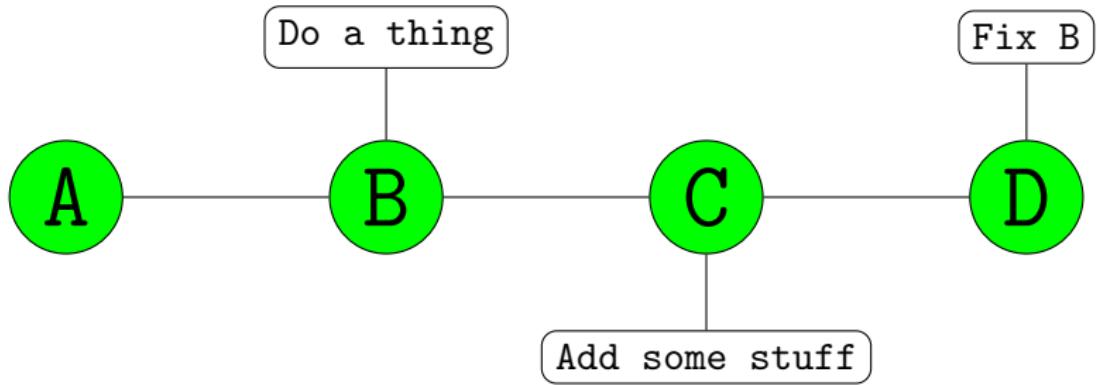
A

B

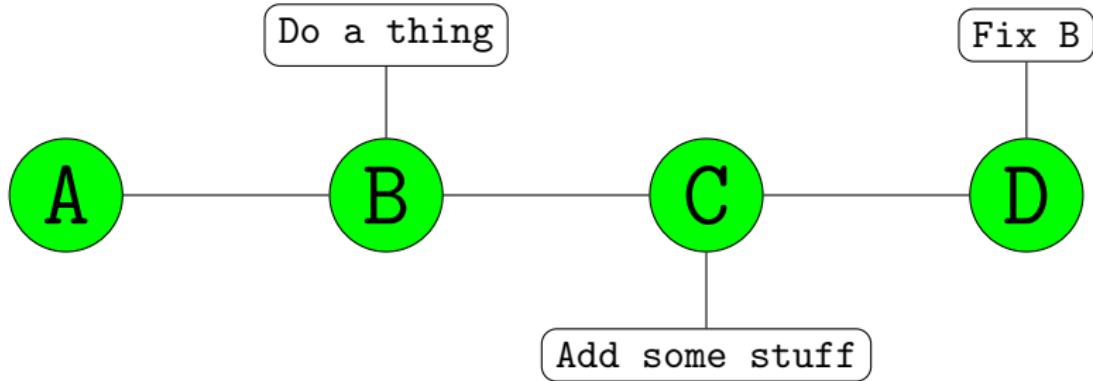
C



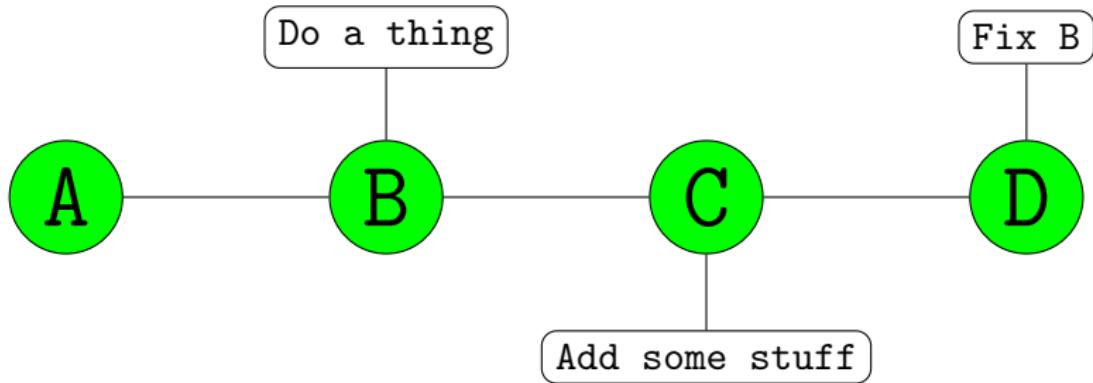




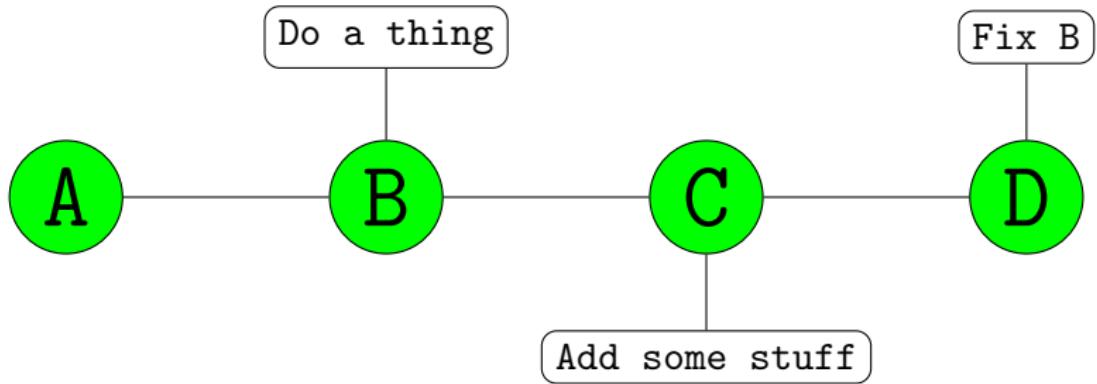
```
git rebase -i A
```



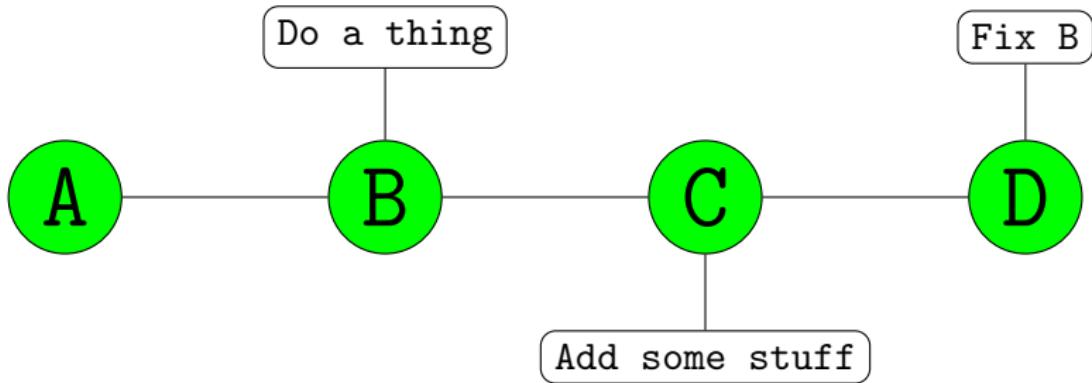
pick B - Do a thing  
pick C - Add some stuff  
pick D - Fix B

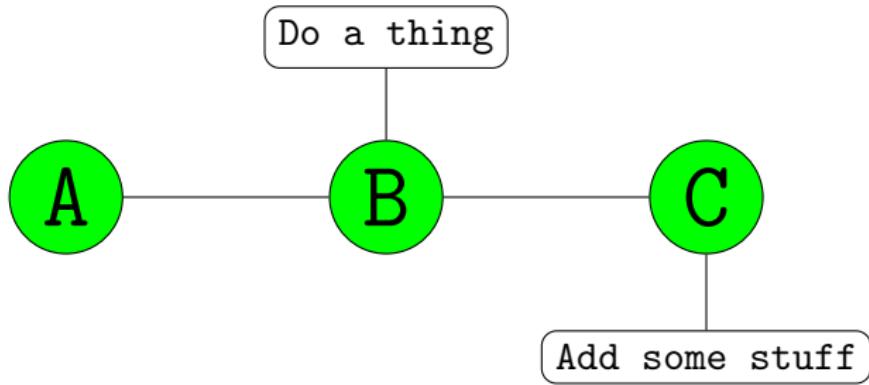


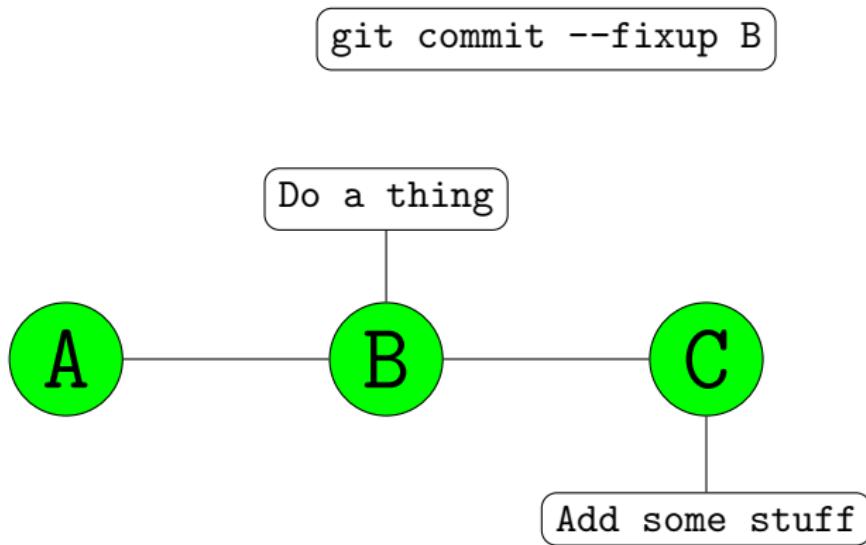
pick B - Do a thing  
pick D - Fix B  
pick C - Add some stuff

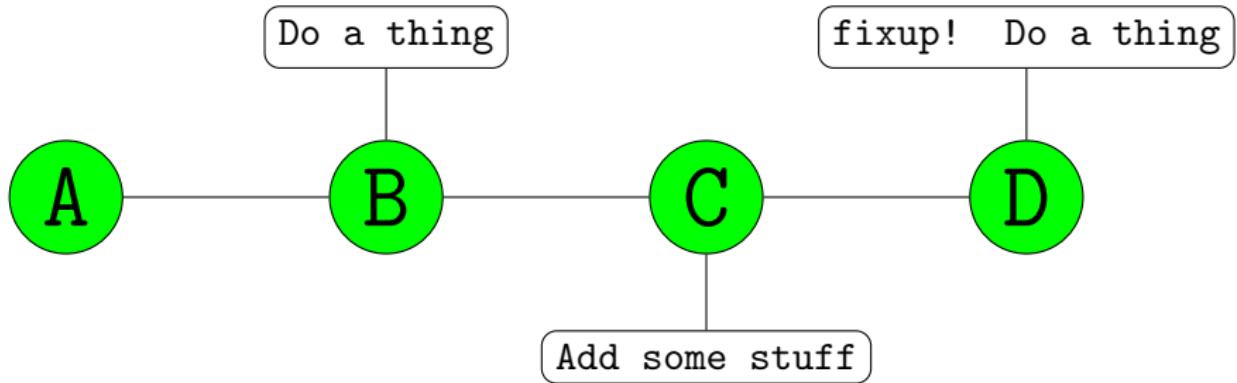


pick B - Do a thing  
fixup D - Fix B  
pick C - Add some stuff

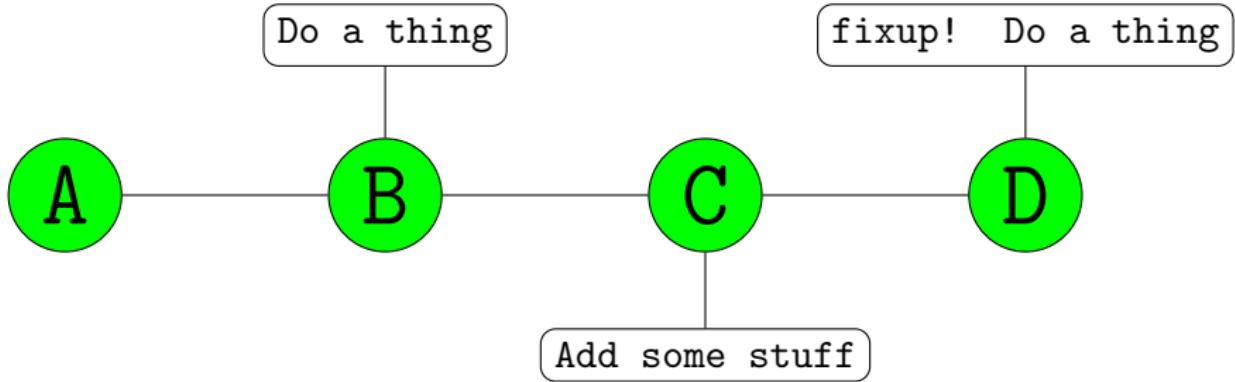




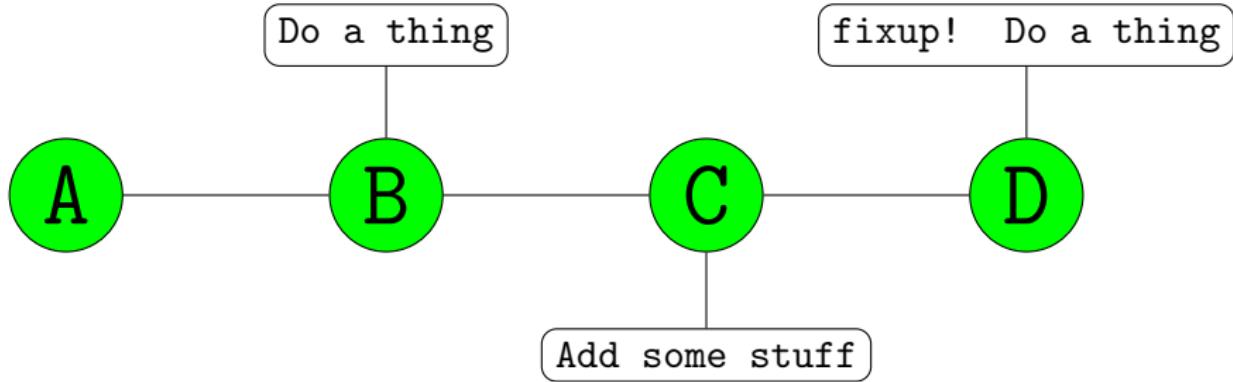




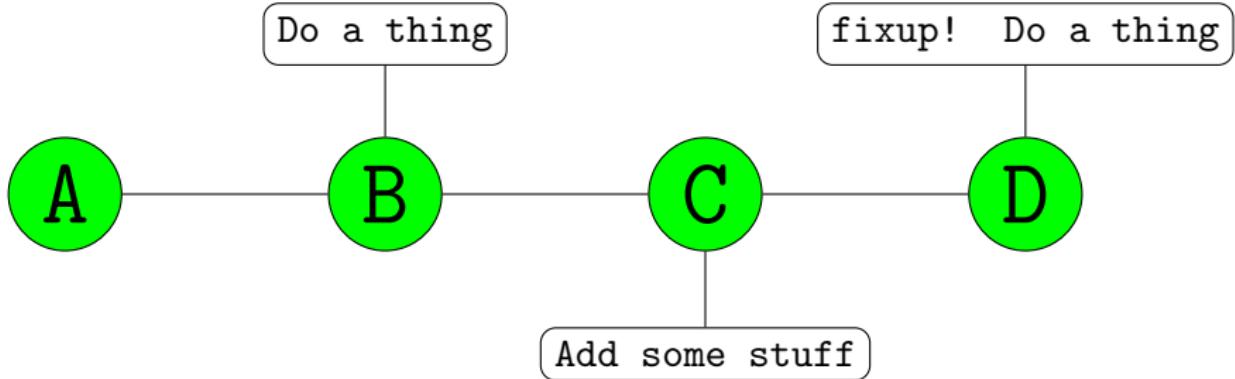
```
git rebase -i --autosquash A
```

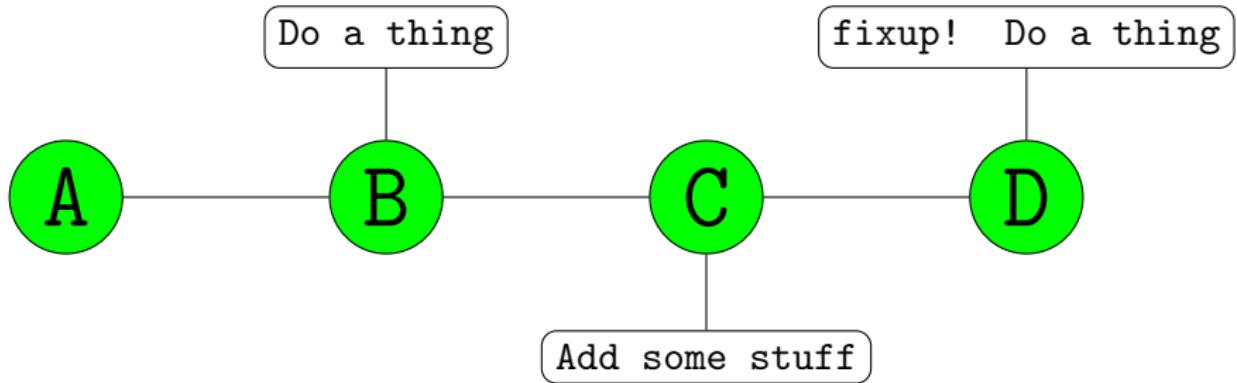


pick B - Do a thing  
fixup D - fixup! Do a thing  
pick C - Add some stuff

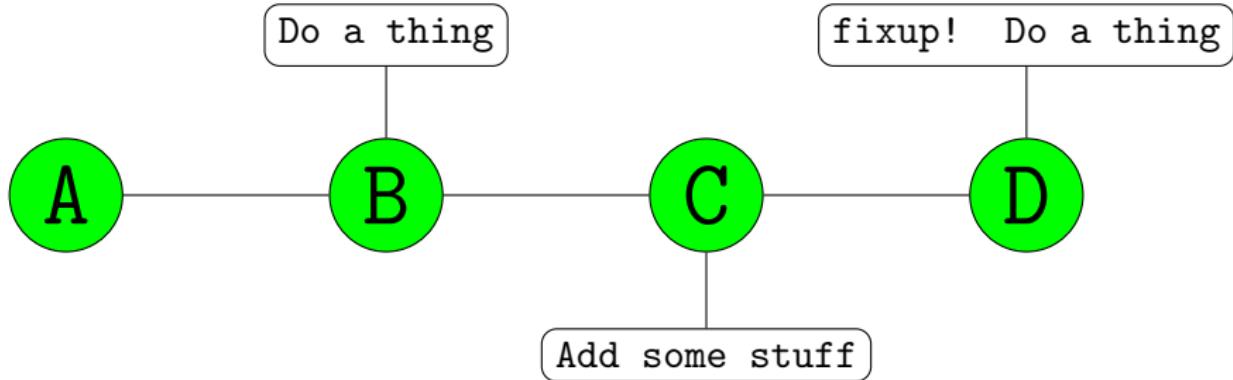


```
git config rebase.autosquash true  
git rebase -i A
```

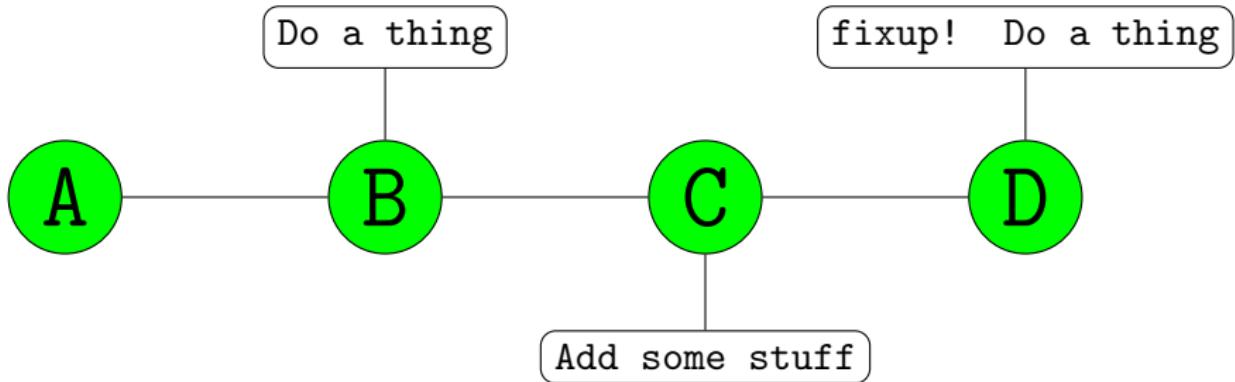




```
git rebase -i --no-autosquash A
```



pick B - Do a thing  
pick C - Add some stuff  
pick D - fixup! Do a thing



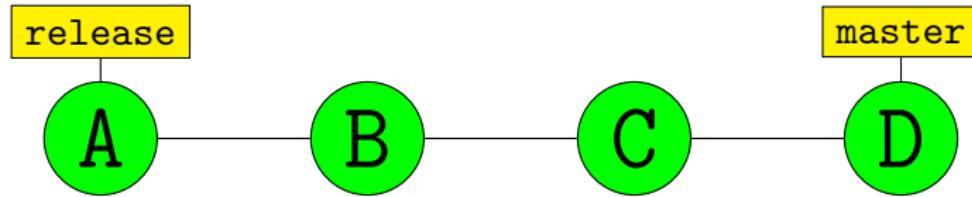
- ▶ Use --fixup commits to “amend” things during regular development.

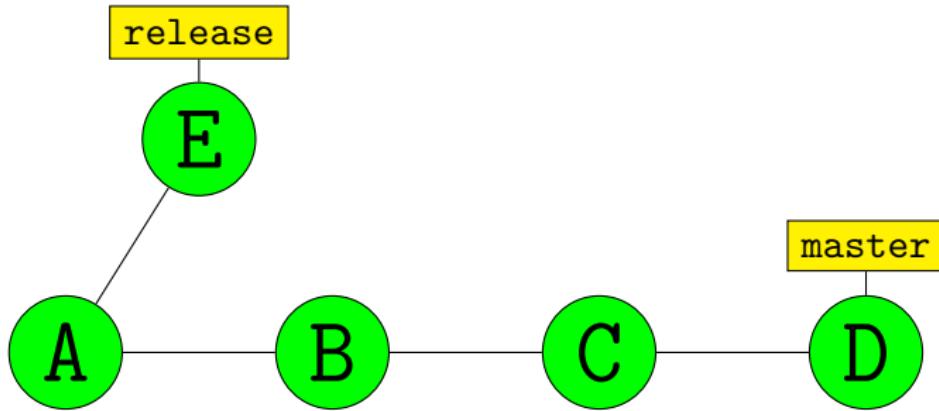
- ▶ Use --fixup commits to “amend” things during regular development.
- ▶ Coordinate with anyone else using the branch before doing a rebase --autosquash.

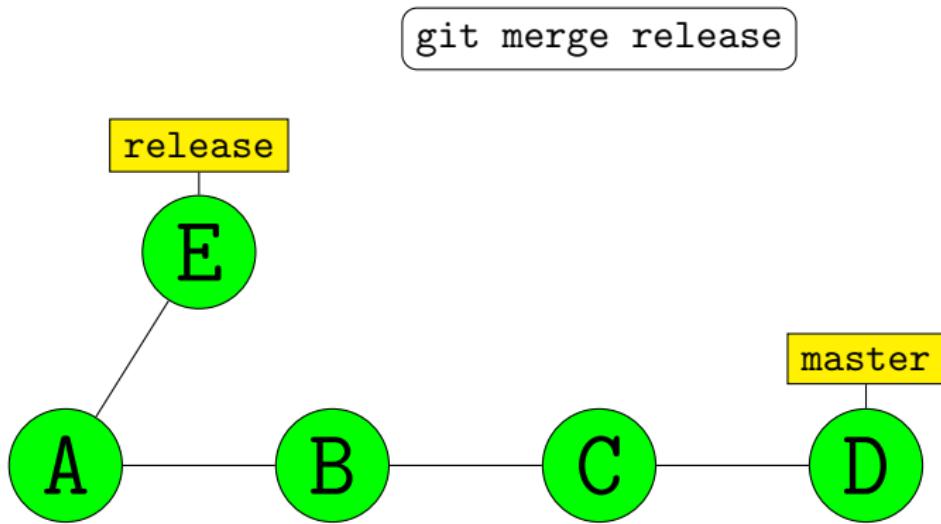
- ▶ Use `--fixup` commits to “amend” things during regular development.
- ▶ Coordinate with anyone else using the branch before doing a `rebase --autosquash`.
- ▶ Don't do it on `master`

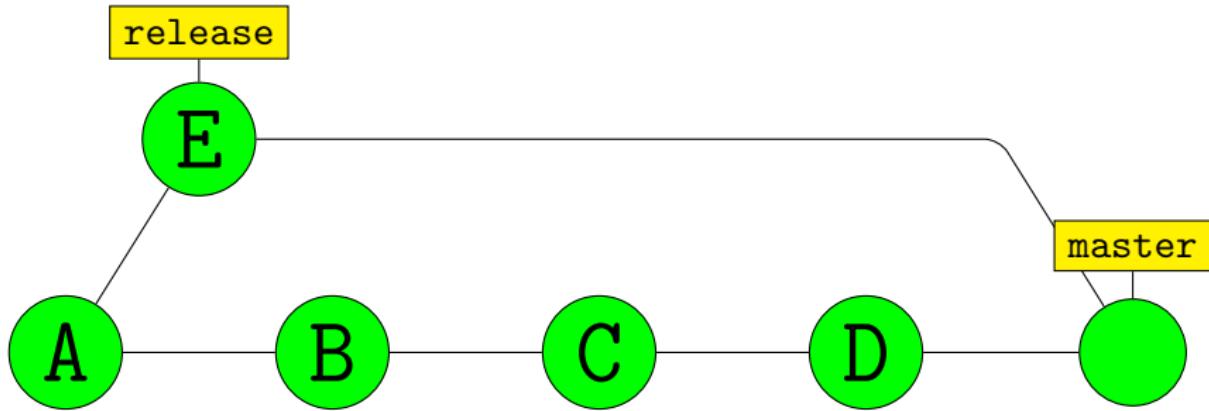
- ▶ Use `--fixup` commits to “amend” things during regular development.
- ▶ Coordinate with anyone else using the branch before doing a `rebase --autosquash`.
- ▶ Don't do it on `master`
- ▶ ... or `release`

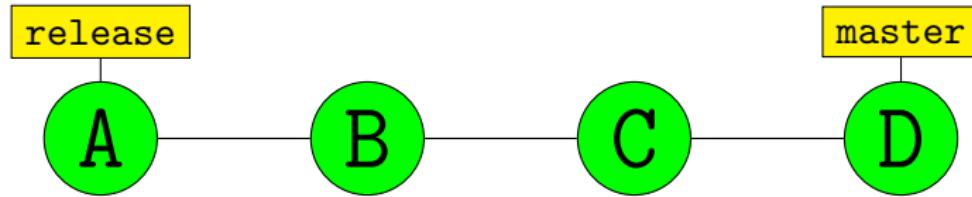
# Merge Down



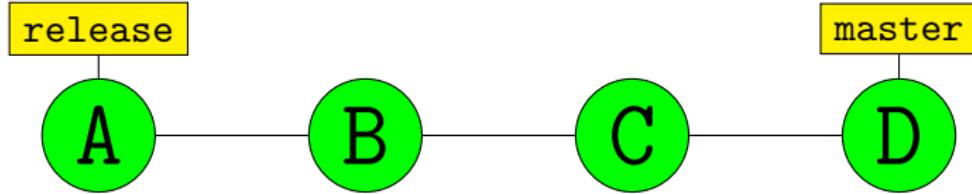


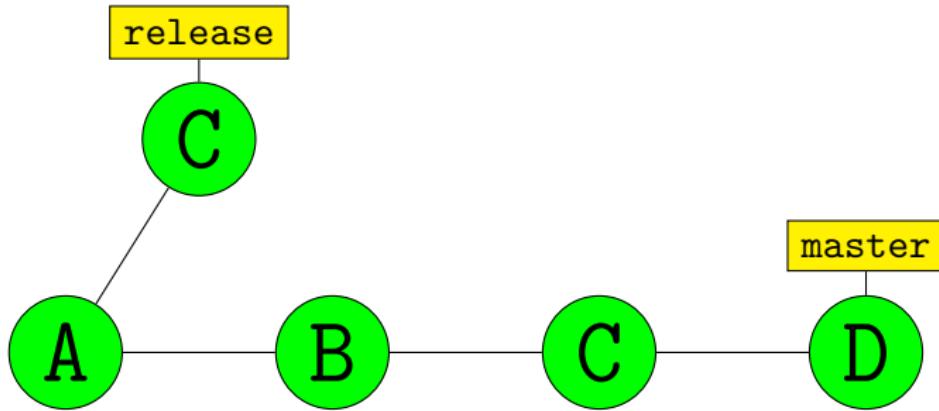


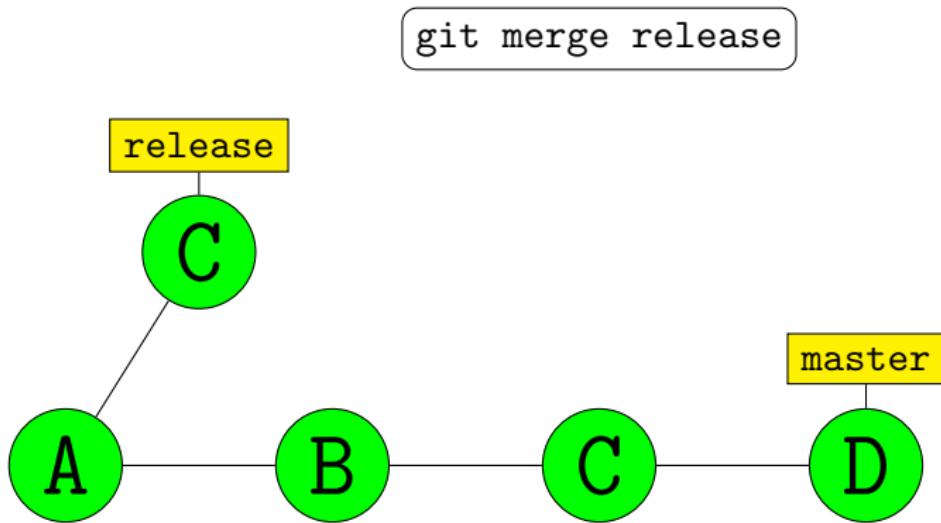


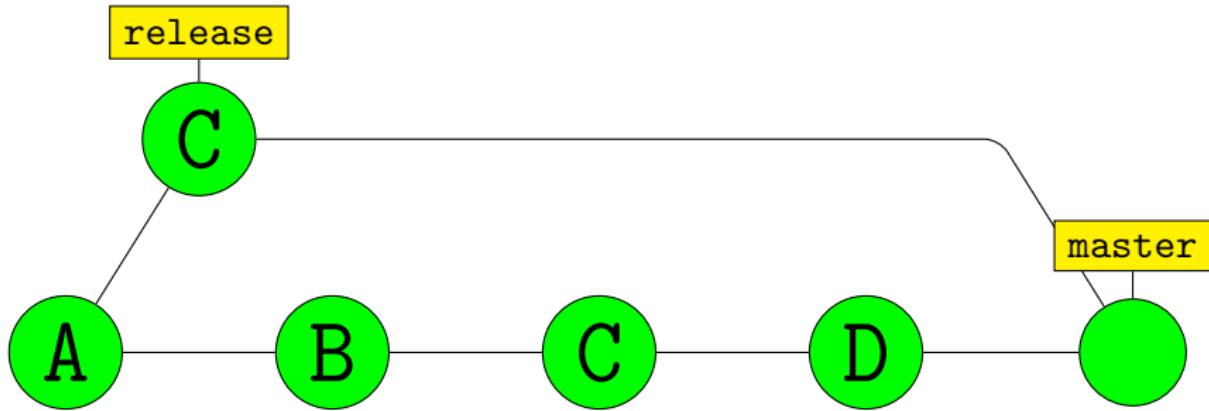


```
git cherry-pick -x C
```









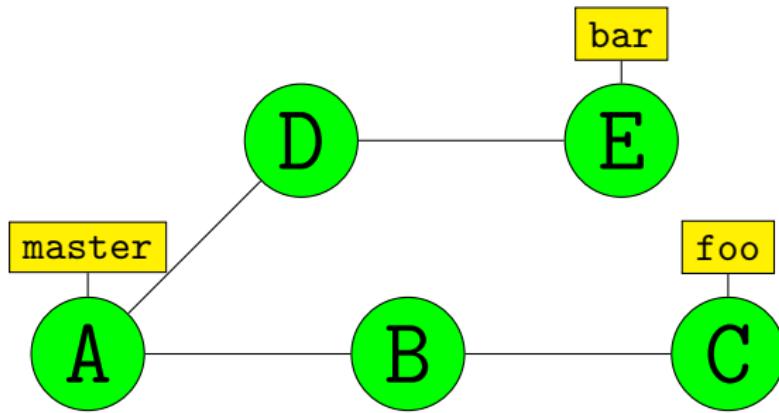
# Log Thoughts

git log is useless

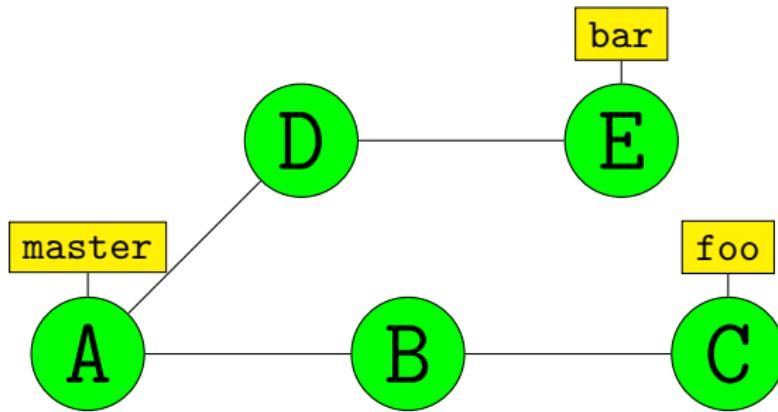
Use git log --graph

Or git hist

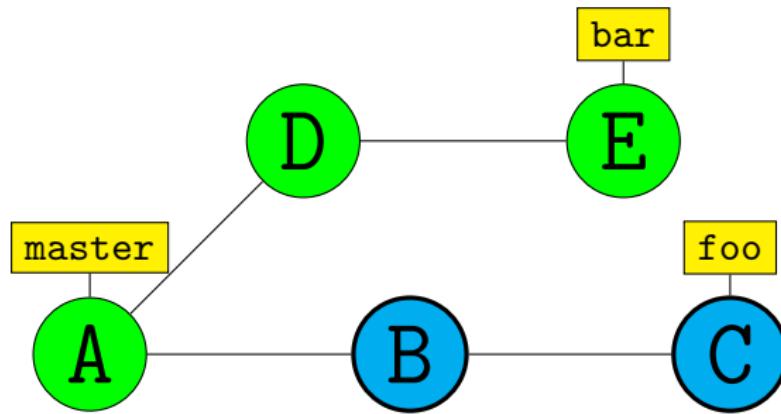
# Diff Tricks



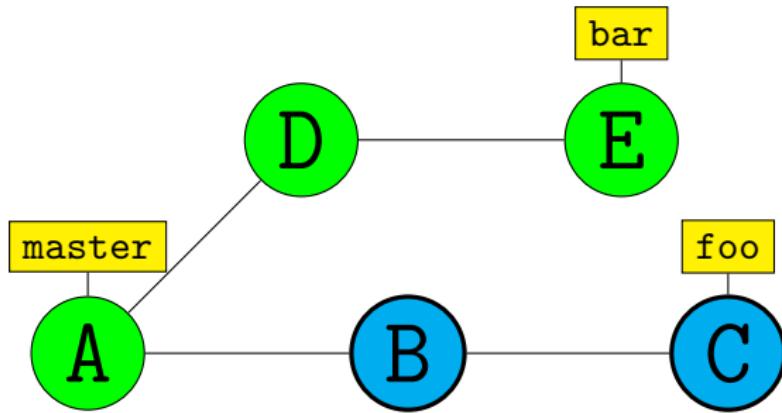
git diff A..C



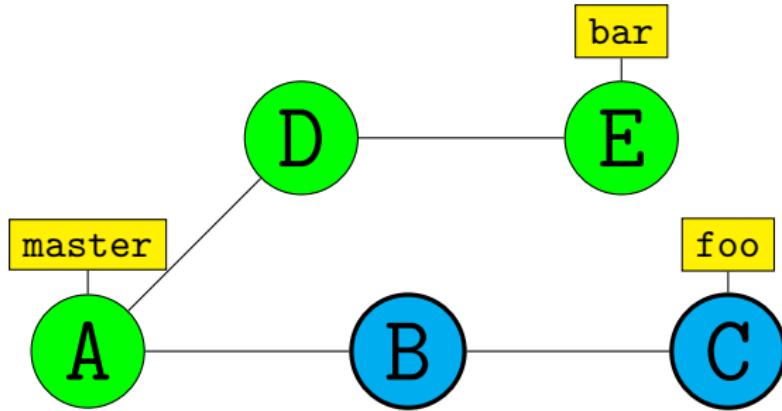
git diff A..C

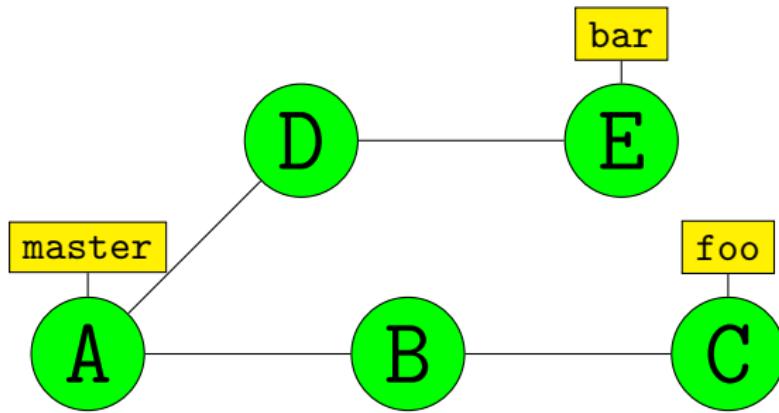


git diff master..foo

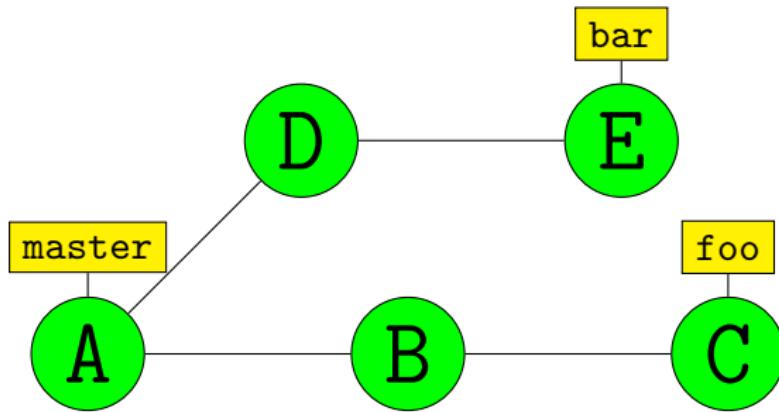


git diff master...foo

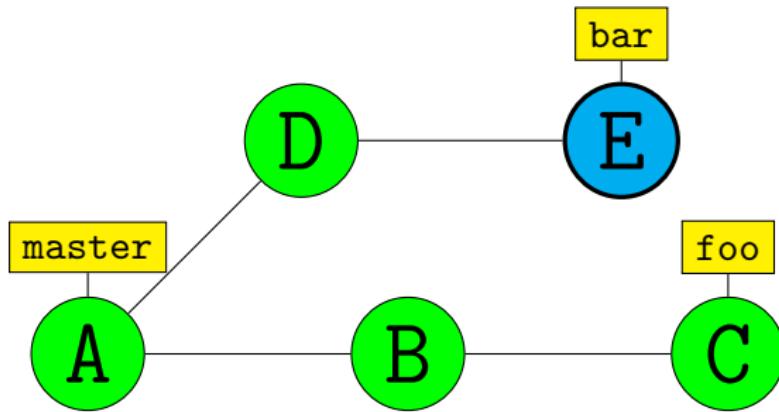


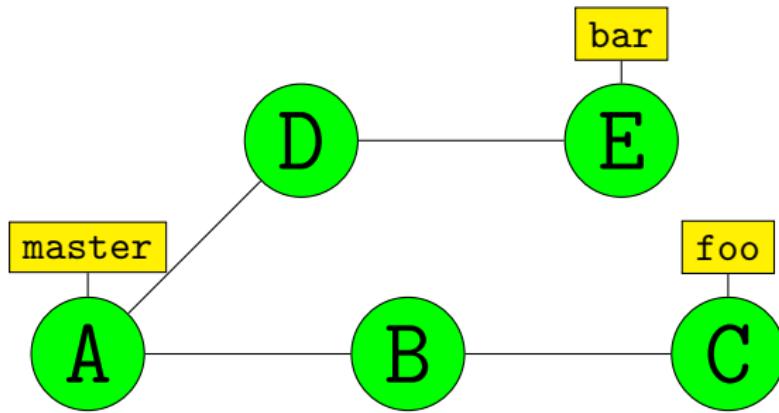


git diff D..E

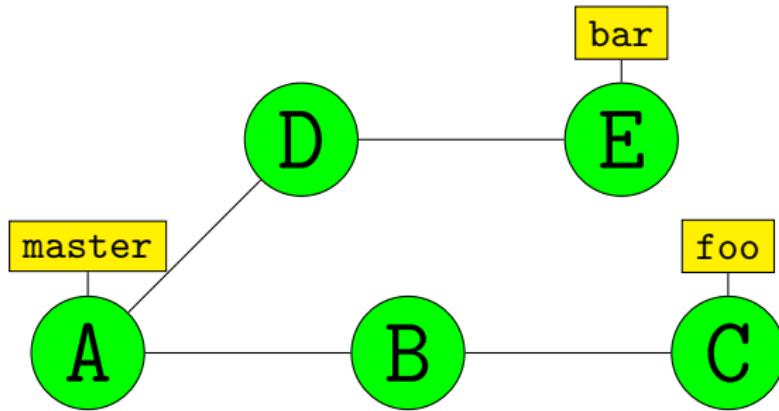


git diff D..E

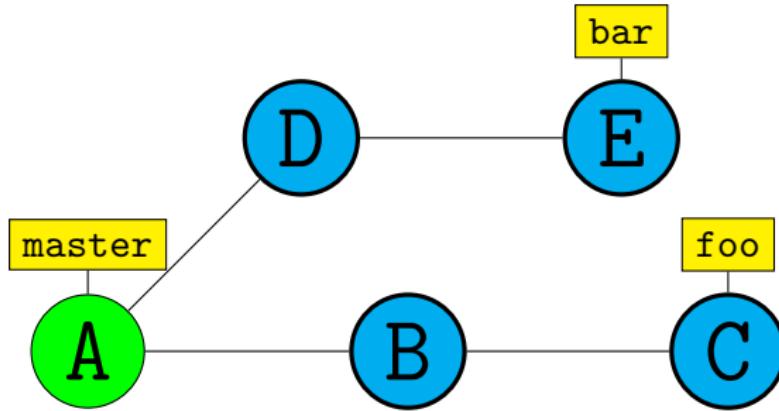


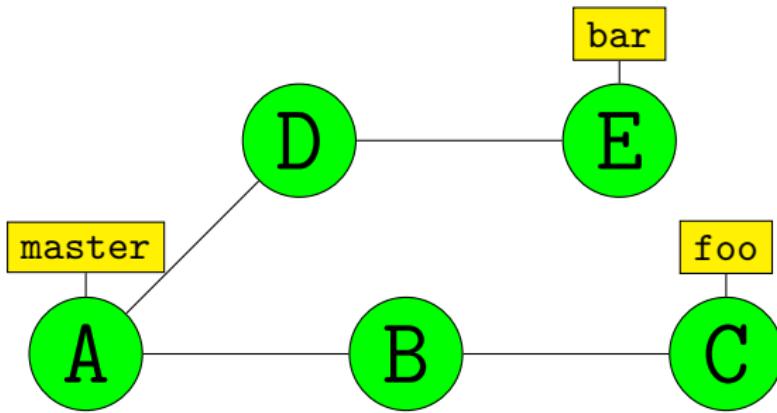


git diff foo..bar

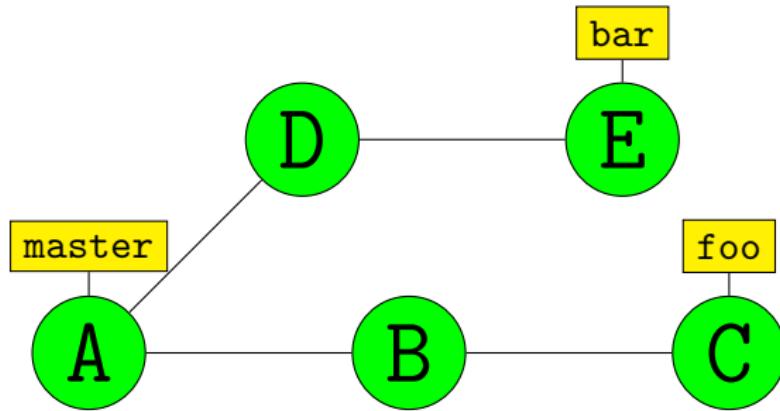


git diff foo..bar

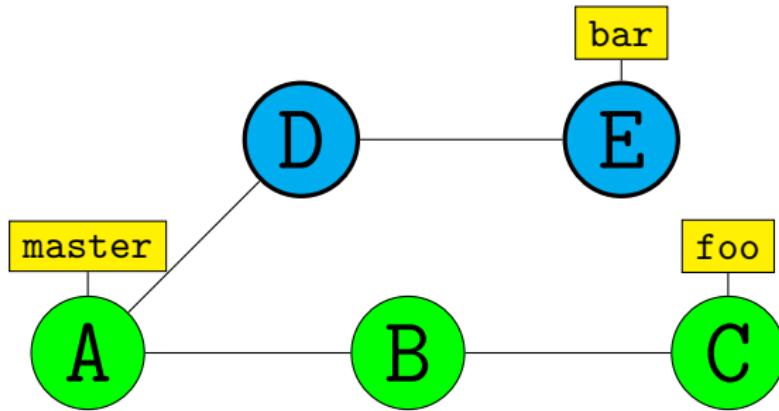


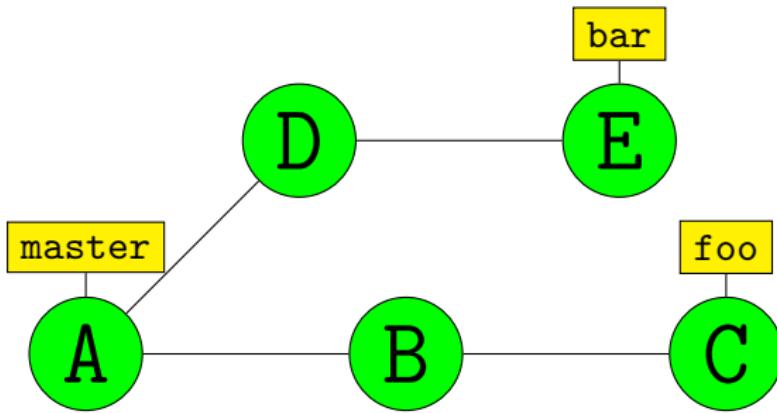


git diff foo...bar

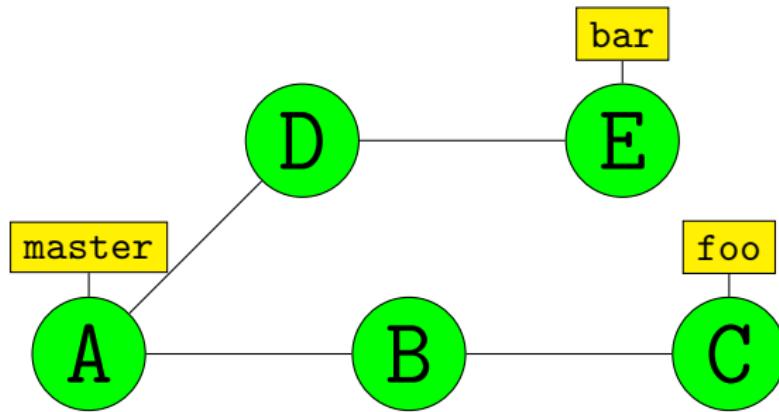


git diff foo...bar

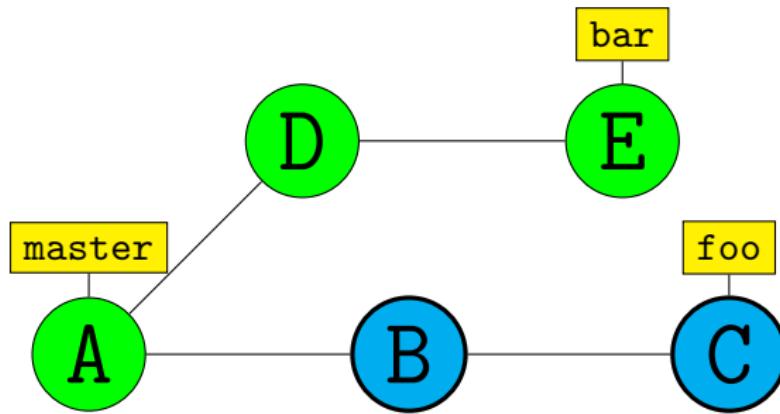


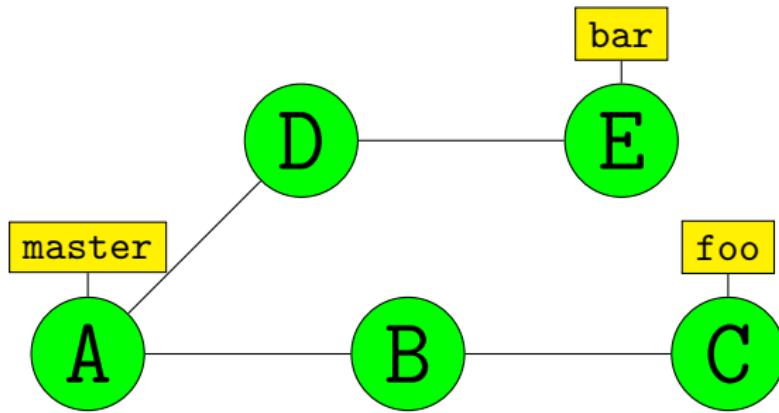


git diff bar...foo

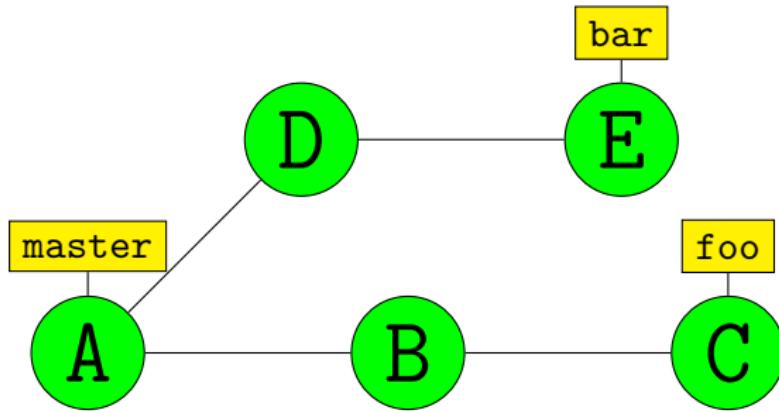


git diff bar...foo

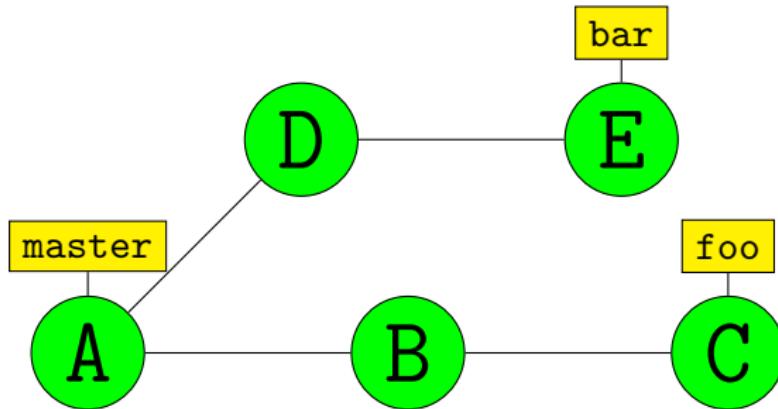




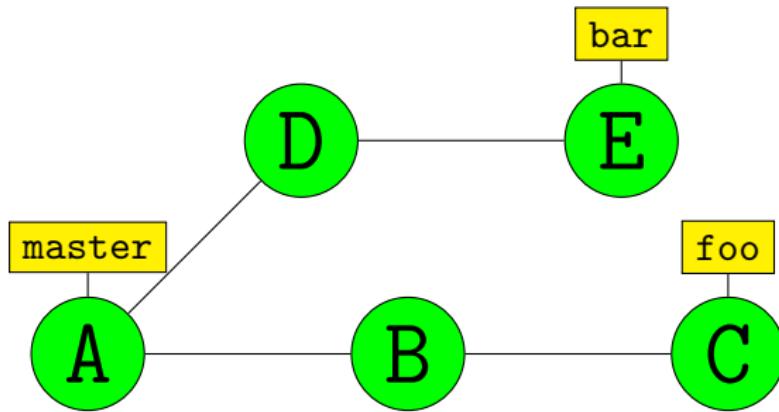
```
git diff master..bar spec
```



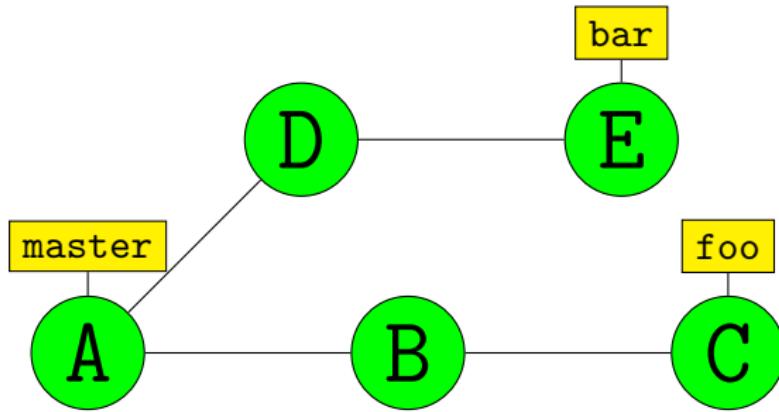
```
git diff master..bar spec | git apply
```

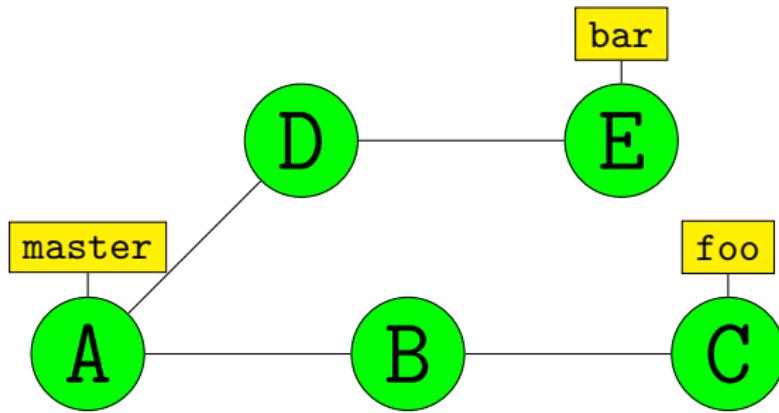


```
git diff master..bar spec > blah.patch
```

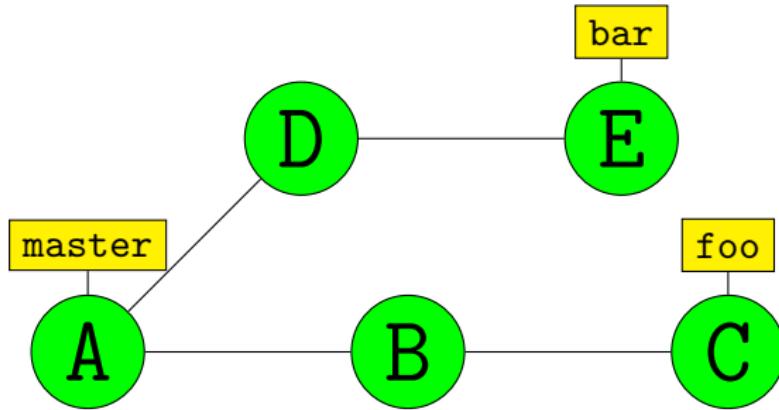


git apply blah.patch

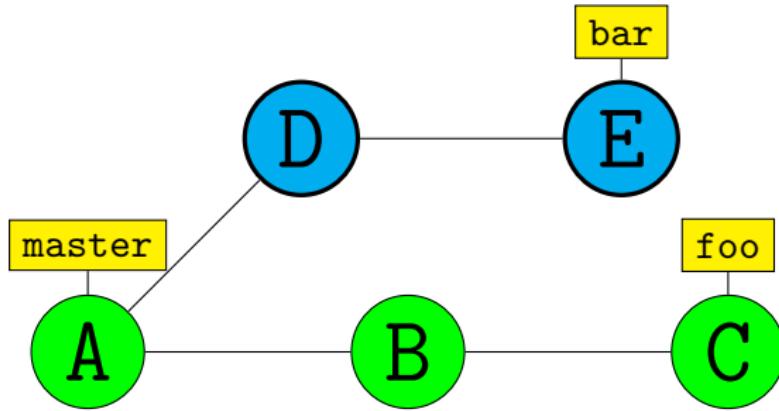


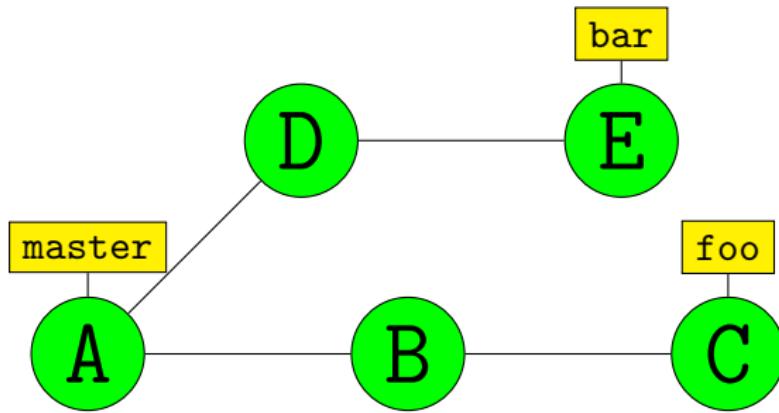


git log foo..bar

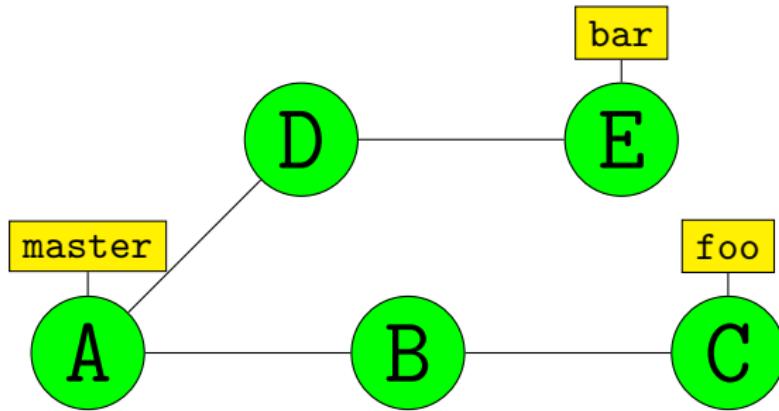


git log foo..bar

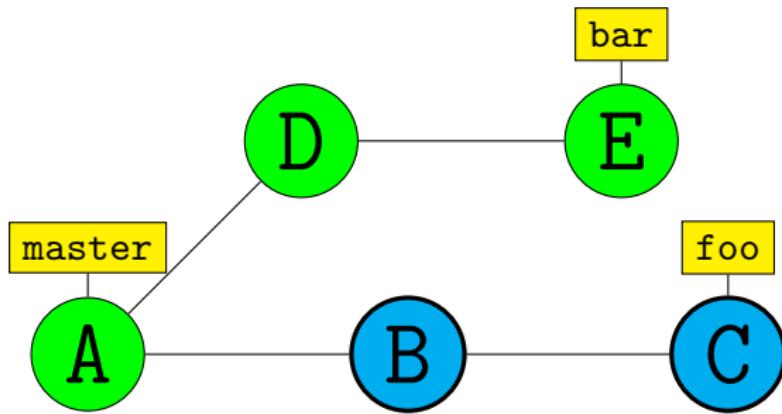


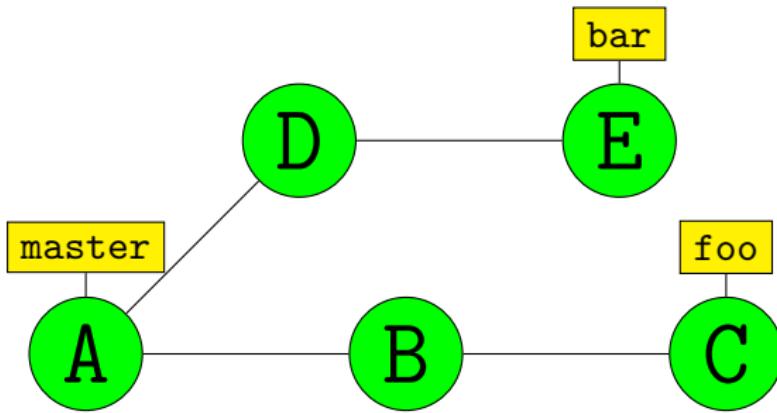


git log bar..foo

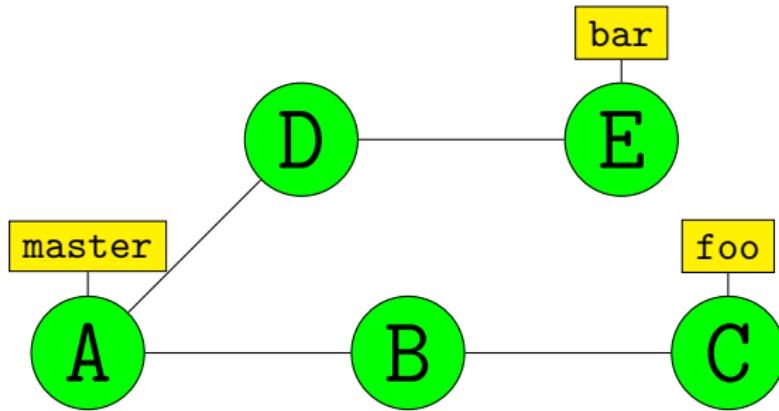


git log bar..foo

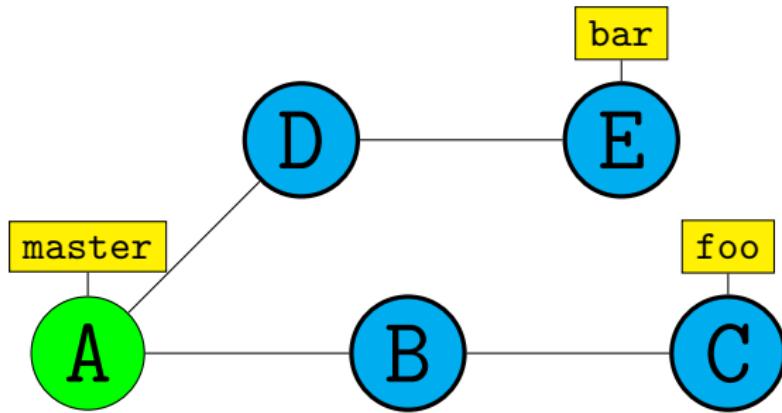




git log foo...bar



git log foo...bar



# Summary

## Summary

- ▶ The goal of Git is to reveal *intent*

## Summary

- ▶ The goal of Git is to reveal *intent*
- ▶ Rewrite history to reveal intent

## Summary

- ▶ The goal of Git is to reveal *intent*
- ▶ Rewrite history to reveal intent
- ▶ Be courteous when doing that on pushed branches (not master or release!)

## Summary

- ▶ The goal of Git is to reveal *intent*
- ▶ Rewrite history to reveal intent
- ▶ Be courteous when doing that on pushed branches (not master or release!)
- ▶ Know what your merge strategy communicates

## Summary

- ▶ The goal of Git is to reveal *intent*
- ▶ Rewrite history to reveal intent
- ▶ Be courteous when doing that on pushed branches (not master or release!)
- ▶ Know what your merge strategy communicates
- ▶ Ancestry matters

# Summary

- ▶ The goal of Git is to reveal *intent*
- ▶ Rewrite history to reveal intent
- ▶ Be courteous when doing that on pushed branches (not master or release!)
- ▶ Know what your merge strategy communicates
- ▶ Ancestry matters
- ▶ Use fixup commits!

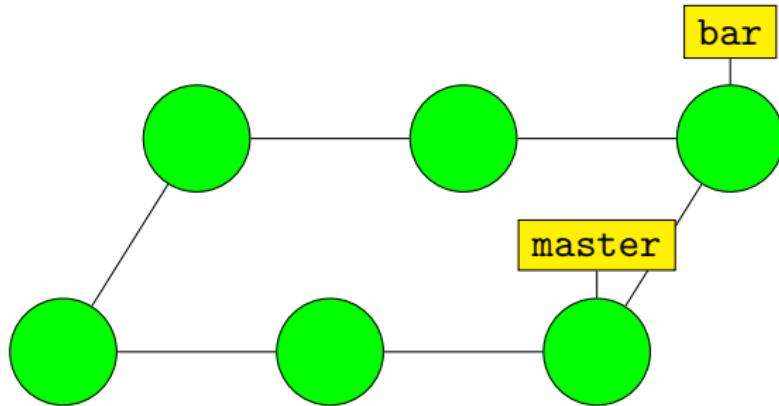
# Summary

- ▶ The goal of Git is to reveal *intent*
- ▶ Rewrite history to reveal intent
- ▶ Be courteous when doing that on pushed branches (not master or release!)
- ▶ Know what your merge strategy communicates
- ▶ Ancestry matters
- ▶ Use fixup commits!
- ▶ Git is awesome!

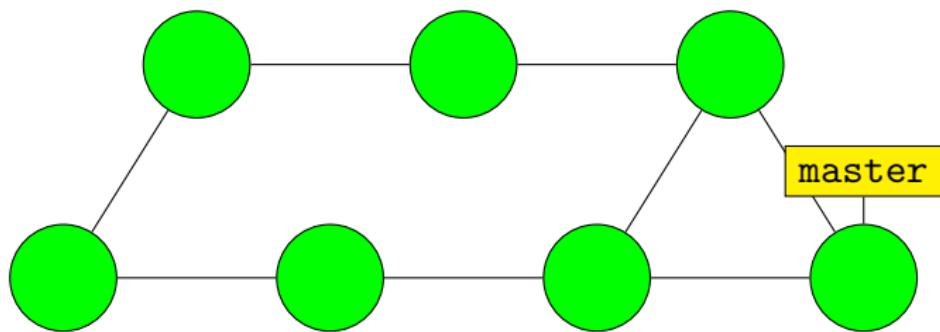
Questions?

Mistake?

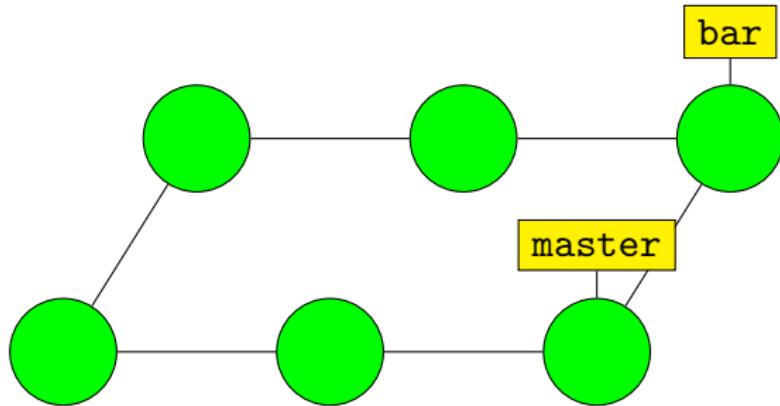
## Merge Strategy



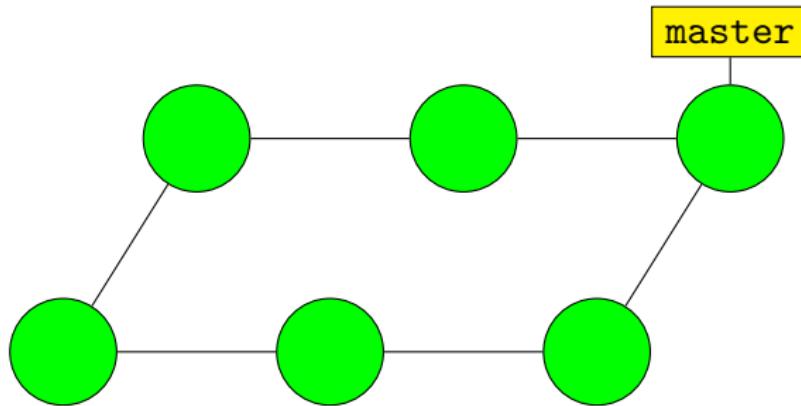
## Merge Strategy



## Merge Strategy



## Merge Strategy



Questions?