Version Control and Git

Linear history - Sequential history of saving on top of a file, when working alone
Multiple authors - Document sent to multiple people, each of who make edits, forming a tree (no cycles, directed edges)
Merging changes - Merging two different versions of an original document into one document

1 Version control software

- RCS (Revision control system) - in order to save space, only stored difference between files, rather than the whole files. Stores latest versions, and stores diffs between previous versions
- CVS (Concurrent Versions System) - Managed under RCS, manages collections of files
- Microsoft word track changes
- Subversion
- git - Doesn't use diff and patch

2 All about git

RCS was designed to be compact, git is designed to be fast

- Distributed version control system - no central repository of anything
- Developed by Linus Torvalds and others to manage the linux kernel
- Designed to be fast
- Very widely used in academia and industry
- Stores every version of every file produced

2.1 Git under the hood

- Different from earlier systems such as RCS - no diffs
- Originally developed under Linux, but available elsewhere
- No central repository, but can synchronise with remotes
- Cloud hosted repository servers: github etc

2.2 Key concepts in git

- A **file** (in a path)
- A **commit**: a snapshot of a collection of files at a particular time
- A **branch**: A linear sequence of commits - always a previous version on a branch, and relation known
- A **repository**: (possibly) many branches of a project
- A **remote**: another place where a repository is stored

2.3 Key commands in git

```
git init
git add
git status
git commit
git push
```

2.3.1 git init

Creates a directory `.git` where everything is stored. You may also want to do `git config` at this stage. Think about adding a `.gitignore` file

2.3.2 git add

Puts current working version of a file into the staging area (area for all files to be committed, copy not cut)
Preparing for a commit
Check what will be committed with git status

2.3.3 git commit

Creates a new commit in the git branch
Makes a commit based on currently staged files
Will start an editor (`git config`)
Consider `git commit -m "message"` to avoid editor

2.3.4 git push

Pushes a branch to a remote repository
`git push origin master`
origin defined by `git remote add origin` or `git clone`

3 Using git

To make a git folder, either use git init to start a new repo, or git clone to copy a repo from a remote repository.

Git add adds the files you want to commit to staging

Git commit will move the files from the staging area to the head of one of the branches in the repository.

Git push moves the files into the remote repository

Git add does not also commit as there are some changed files that you do not want to commit to the repository.

To move a repo to github, create an empty repository and set the remote.