Today's plan

- Class interactions
    - Ex 1-1
    - Keys concepts
    - Recap discussion
- Class exercises
    - Ex 1-2

Ex 1-1

Any volunteer?

Key points - editors

- command line editors: **emacs** vs **vim**
- save and quit: ‘Ctrl-x Ctrl-s’, ‘Ctrl-x Ctrl-c’ vs ‘Esc :xq’
- quite w/o saving: ‘Ctrl-x Ctrl-c’, ‘no’, ‘yes’ vs ‘Esc :q!’
- undo and redo: ‘Ctrl-x u / Ctrl-x U’ vs ‘Esc :undo / Esc :redo’
- for/backward search: ‘Ctrl-s/Ctrl-r word’ vs ‘Esc /word’, ’n/N’
- find and replace: ‘Esc % word’, ‘new word’, ‘n’ for next, ‘ ’ for replace vs ‘Esc :%s/word/new word/gc’, ‘n’ for next, ‘y’ for replace, ‘a’ for replace all
- go to a line: ‘Esc g g’, ‘line number’ vs ‘Esc line number G’
- show line numbers: ‘Esc linum-mode’ vs ‘:set (no)number’
- always show line numbers: edit ∼/.emacs.d/init.el’, add ‘(global-linum-mode)’ vs edit ‘∼/.vimrc’, add ‘set number’

Key points - Git

- Why do we use Git? a powerful version control tool
- Repository: a place that stores (snapshots of) our codes
- Remote vs local
  - First, update the local repo from the remote one (pull)
  - Then, make changes in the local repository (add, commit per each change - don't make lots of changes, a tiny step each time)
  - Last, push the changes to the remote repo (pull, push - need to pull it again to check if there are more updates since the last pull)
- clone a remote repo to a local repo: **git clone git/https link**
- get update from remote repo: **git pull**
- add changed files to a commit: **git add files**
- commit a change: **git commit -m comments**
- push change to remote repo: **git push**

Key points - Git

- stash a change (current work): **git stash**
- pop back the stash: **git stash pop**
- discard the stash: **git stash clear**
- reset local to last commit: **git checkout .**
- reset files to last commit: **git checkout files**
- reset local to a commit: **git checkout commit id**
- revert a commit: **git revert commit id, git commit**
- revert last two commits: **git revert HEAD~2..HEAD, git commit**
- list of commits: **git log**
- show changes before git add: **git diff files**
- show changes before git commit: **git diff HEAD**
- show changes between commits: **git diff commit1 commit2**

Key points - submission workflow

- You can develop your codes anywhere you want, but
- you must test your codes on the ugrad server
- Put your codes to ugrad
  - use **scp/pscp** to copy from your local machine to ugrad server
  - Or, use **git** to create another local repo on ugrad server
- After testing your codes, submit a **zip** file includes
  - your source codes
  - gitlog.txt
  - README (not asked for hw0, but usually required)
- Do not submit with the followings
  - executable
  - object files (.o files)
- Submit the zip file to GradeScope. Submit it whenever you finish a task. Never wait until the last moment to submit.

Why do we use version control system like Git?

Mentimeter

So that we can revert to a previous change if the current version doesn't work

So that you can always go back and undo changes if a change doesn’t work

Version control allows you to record snapshots of your project

allow multiple people to work on a large project

- Work in teams,- Provide a history of changes,- Redundancy

work remotely

It makes it very easy to share code and view changes over time.

Because it is a powerful way to organize working on code projects in big teams with multiple members

To be able to continuously save our code and collaborate

18

Why do we use version control system like Git?

Mentimeter

To track changes we make on files and go back to check work history

Because it makes it easy to keep track of what one's changed and to revert when a bug orrur.

Provides a history of previous edits, allows for remote work and team collaboration

To act as a shared "drive" for multiple people to work on codes together, as well as provide an organized system in case errors happen

It is a good way to share files

So that you can go back to a previous version if you want

To collaborate with others on programming projects

To keep track of changes

It is more efficient

18

Mentimeter

Name six common Git commands.

git add filename

git log

git commit

git push

git clone

pull

git pull

init

git diff files

log

git

push

add

git checkout

stash

clone

commit

git stash

git commit -m

fetch

status

diff

git add

git commie

git add file

19

Mentimeter

What are the files that must be included in your submission?

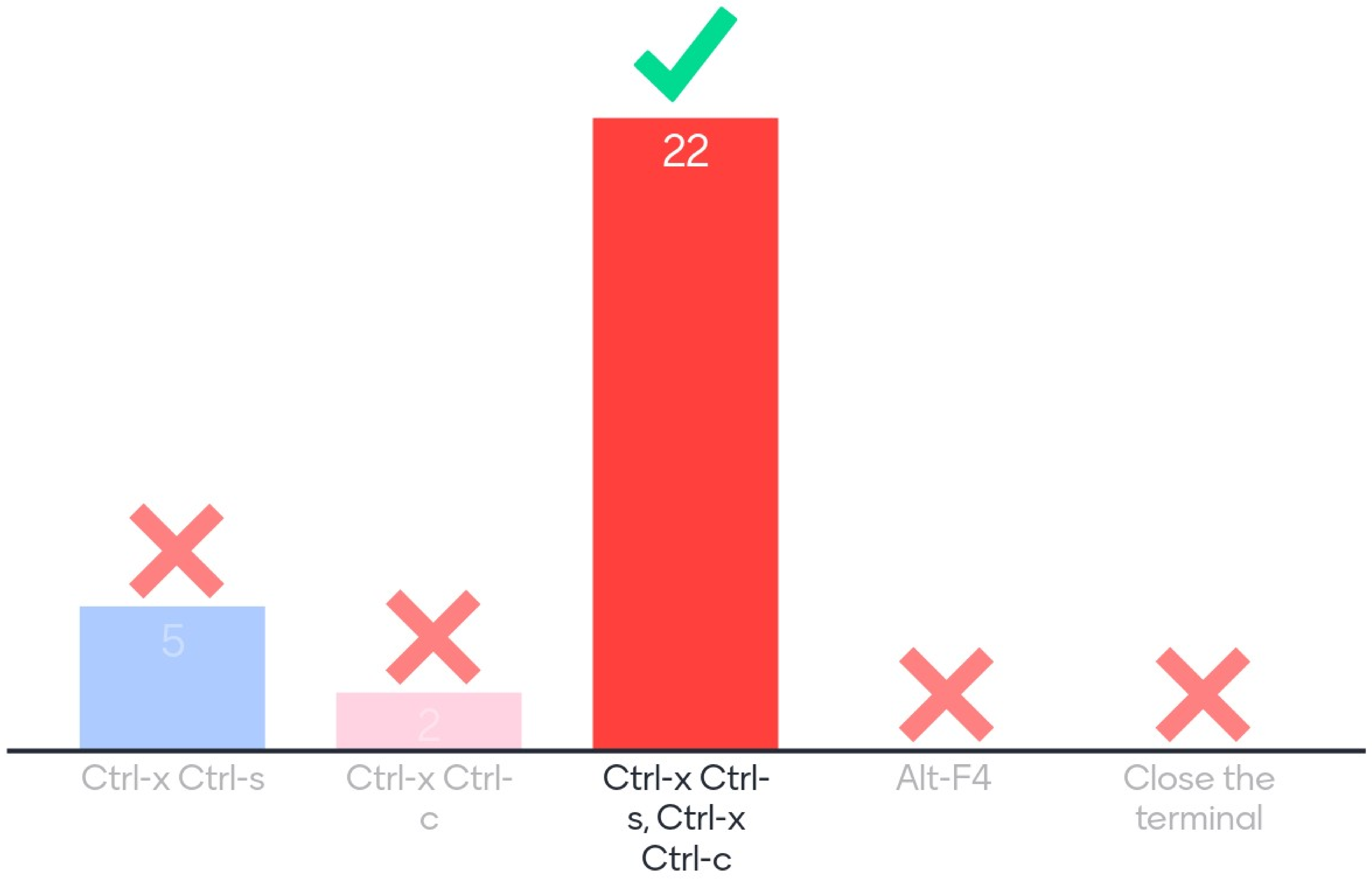
git log ✗ 2x	gitlog .txt source codes README ✗	gitlog, C source files, README ✗
zip file with .c and gitlog.txt ✗	README, gitlog.txt, source code files ✗	source code, gitlog.txt ✗
code (.c files) gitlog.txt, zip.zip ✗	readme, source code ✗	source code, commit log, ✗

The correct answer is: source codes, gitlog.txt, README (not for hw0)

28

How do you save and quit on emacs editor?

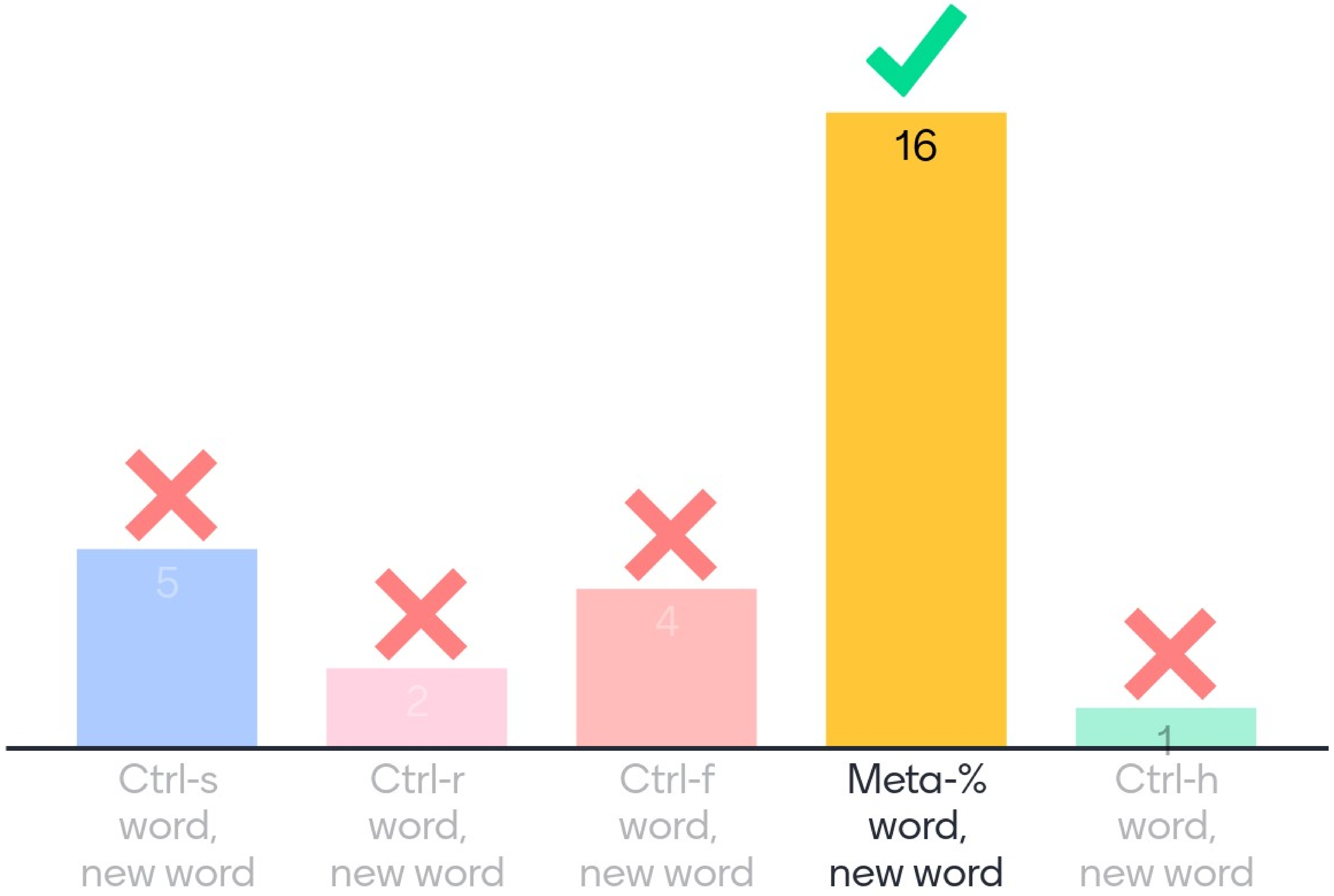
Mentimeter



29

How do you search and replace on emacs?

Mentimeter



28

Class exercises

Ex1-2 and Ex1-3