CS 35L Fall 19 Section 3 Notes W2 Wed
Zhaowei Tan

Regex:
Refer to the cheat sheet at:
http://www.regexlib.com/CheatSheet.aspx?AspxAutoDetectCookieSupport=1

Play around with this online tool:
https://www.regexpal.com/

Some highlights:

1. (Anchors) ^ start of a line, $ end of a line
2. (Character classes) . means any character except newline
   [abc] any of a, b, or c (only matches one occurrence)
   Or we can use [a-c], [a-g1-9] to represent a range of char/number
   We also have \w, \d, ... for word, digit, ...
3. (Quantification) After an expression (a single character class or a group), we can use *, + or ? to represent multiple occurrence of the expression
   * zero or multiple occurrence
   + one or more occurrence
   ? zero or one occurrence
   E.g. a*b will match b ; a+b will not match b
4. {} used to specify certain times of the occurrence (compared to * and +, which allows arbitrary many)
   E.g. [abc]{3} will match aaa but not bb
   [abc]{1,3} matches a string with 1-3 occurrence of [abc].
5. \ escape to escape special characters
6. (Grouping) () parenthesis used to capture group
   E.g. (abc){2} will match abcabc
7. Lazy vs. greedy, lookaround, back-reference: see slides
8. sed command:
   Substituting text:
   sed 's/regex/replacement/flags' file
   E.g. sed 's/^a/b/' file.txt
   sed 's/a$/b/g' file.txt   (-g flag: substitute all the occurrence)
   The delimiter for sed can be another sign, e.g: sed 's_a_b_g'
9. To re-use the searched regex pattern use & sign. E.g. if you want to wrap every matching substring with a pair of parenthesis, do this: sed 's/ab+/(&)/g'
10. To delete with sed, use sed '/pattern to match/d'
11. Awk basics: https://www.geeksforgeeks.org/awk-command-unixlinux-examples/
12. Extended vs. basic regular expression. Some special signs need escape to have special meaning:
    https://www.gnu.org/software/sed/manual/html_node/BRE-vs-ERE.html