sort, uniq, cut and tr

Slice and count text

CONCEPTS

These are line filters: each reads stdin or files and writes stdout, so they chain with pipes. tr reads stdin only.

cut pick columns
↓ lines
sort order / group
↓ adjacent
uniq collapse, count
↓ counts
sort -rn rank by count

uniq only merges adjacent lines, so sort first (or use sort -u).

Sort order depends on the locale. Use LC_ALL=C for plain byte order, and it is also much faster.

SORTING

`sort <file>`
Sort lines alphabetically

`sort -r <file>`
Reverse the order

`sort -n <file>`
Numeric sort

`sort -h <file>`
Human sizes such as 2K, 1G

`sort -u <file>`
Keep one of each equal line

`sort -V <file>`
Version sort (1.2 before 1.10)

SORTING BY KEY

`sort -k <n> <file>`
Sort from field n to end of line

`sort -k <n>,<n> <file>`
Sort on field n only

`sort -t <sep> -k <n> <file>`
Set the field separator

`sort -k <n>n,<n> <file>`
Numeric on one field

`sort -k <a>,<a> -k <b>nr,<b> <file>`
Primary then secondary key

UNIQ

`uniq <file>`
Collapse adjacent duplicates

`uniq -c <file>`
Prefix each line with its count

`uniq -d <file>`
Only repeated lines, one each

`uniq -u <file>`
Only lines that never repeat

`sort <file> | uniq -c | sort -rn`
Rank lines by frequency

CUT

`cut -f <list> <file>`
Select tab-separated fields

`cut -d <sep> -f <list> <file>`
Fields with a custom delimiter

`cut -c <list> <file>`
Select by character position

`cut -f 2- <file>`
From field 2 to the end

cut cannot reorder columns and treats every delimiter separately, so runs of spaces give empty fields. Use awk for those.

TR

`tr <set1> <set2>`
Translate characters one to one

`tr a-z A-Z`
Lowercase to uppercase

`tr -d <set>`
Delete these characters

`tr -d '\r'`
Strip carriage returns

`tr -s <set>`
Squeeze repeats into one

`tr -s ' '`
Collapse runs of spaces

`tr '\n' ' '`
Join lines with spaces

tr works on single characters, not words or patterns. For strings or regexes use sed.