# 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)

`sort -f <file>`
Ignore case

`sort -b <file>`
Ignore leading blanks

`sort -s <file>`
Stable: keep input order on ties

`sort -o <file> <file>`
Write result in place

## 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

## SORTING BY KEY (cont.)

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

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

`sort -t$'\t' -k <n> <file>`
Tab-separated (bash quoting)

Key options like n or r placed on a -k override the global ones for that key only.

## 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

`uniq -i <file>`
Ignore case

`uniq -f <n> <file>`
Skip first n fields when comparing

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

`sort <a> <b> | uniq -d`
Lines present in both sorted lists

## 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 -f <list> --complement <file>`
All fields except these

`cut -s -f <list> <file>`
Skip lines with no delimiter

`cut --output-delimiter=<str> -f <list>`
Change the output separator

## CUT (cont.)

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 '[:lower:]' '[:upper:]'`
Case change using classes

`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 -cd '[:alnum:]'`
Keep only letters and digits

`tr -c <set1> <set2>`
Replace everything not in the set

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

`tr ',' '\n'`
Split on commas into lines

`tr -cs '[:alpha:]' '\n'`
One word per line

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

### Common tr classes

| Class | Matches |
|---|---|
| `[:alpha:]` | letters |
| `[:digit:]` | digits |
| `[:alnum:]` | letters and digits |
| `[:space:]` | all whitespace |
| `[:blank:]` | space and tab |
| `[:punct:]` | punctuation |
| `[:upper:] [:lower:]` | case pairs |

## ALSO HANDY

`paste <a> <b>`
Join files side by side with tabs

`paste -sd, <file>`
Join all lines with commas