Quick Grep

TL;DR : A tiny (<500 loc) C file that does a fast grep with great defaults for programming.

🔍💨✨

Usage

Compile the C file, name the output `gg` (for quick typing) and put it somewhere in your path. It will do a recursive, case insensitive search by default and put in line numbers so the output plays well with many editors like vim.

```
$> grep -inR "int i" .
vs
$> gg int i
```

Vim Usage

Add something like this to your `.vimrc` :

```
command! -nargs=+ Find cexpr system('gg ' . shellescape('<args>'))
```

And you can now use:

```
:Find int i
```

to quickly find relevant results from your current directory downward.

Smart Case

`gg` by default will perform a case-insensitive search but if you provide what looks like capital letters then it will perform a case-sensitive search. This is inspired by the `smartcase` option of vim and works really well.

```
$> gg test     # matches test, TEST, tEsT...
$> gg Test     # matches only Test
```

Run Directory

`gg` searches in current directory but you can pass `-c ../some/other/path` to run the search in another directory.

```
$> gg test                  # searches in current directory and sub-directories
$> gg -c ../some/path test  # searches in ../some/path and sub-directories
```

Invert Results

It's also useful to search for lines that do *not* match a certain condition.

```
$> gg test      # matches test, TEST, tEst... etc
$> gg -v test   # returns lines that do NOT match test, TEST, tEst... etc
```

Performance

`gg` ignores directories like `.git`, or `node_modules`, or `target`, large files, and binary files, and this gives us quite a boost in many cases:

```
$> time grep -inR "int i" . > /dev/null
real    0m32.282s
user    0m15.777s
sys     0m1.600s

$> time gg int i > /dev/null
real    0m1.415s
user    0m0.603s
sys     0m0.139s
```

Of course these are benchmarks on my machine for my specific use case so take them with a large grain of salt. Still, it’s pretty fast for a tiny C script.

Customisation

`gg` gives you zero customisation options. However, because it’s a tiny, simple C file, you can probably make it do whatever you like!

Enjoy!