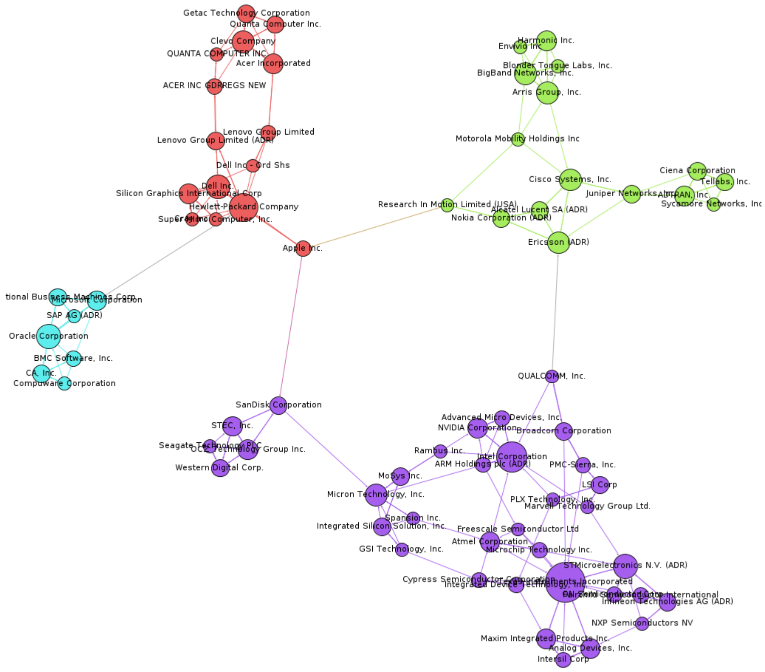
Company was chosen, ex. IBM.
Script was written to extract relations from finance.google.com.
Script is intended to run many times, so it caches request results in files.
Traverse depth and relation count per company can be tuned.
With 3 relations network stops expanding at depth 16 giving 73 nodes.

Graph was visualized in Gephi.
Layout "Force Atlas" was applied.
Graph was partitioned by color (statistics: modularity: resolution 5).
It looks like companies form groups by industry:
computers, software, telecom, chip.
Size of nodes was set (ranking) according to their degree.
This looks like showing main companies inside groups:
TI and Intel, HP, MS and Oracle.
However on setting size by betweeness centrality (statistics: avg. path length)
Apple inc. become the largest integrator.

No sensible way to handle this network data in PAST was found.



See perl sources at https://github.com/rewlad/ttu_andm

```

#perl geta.pl > ibm.csv
use strict;
use JSON::XS;
use List::MoreUtils qw(uniq);

my $depth = 16;
my $relcnt = 3;

sub get{
    my $q = $_[0]=~/^([\w%]+)$/ ? $1 : die;
    my $fn = "cache/$q";
    -e $fn or
        system ’wget’,’-O’,$fn,"http://www.google.com/finance?q=$q"
        and die;
    my $res;
    open FF,’<’,$fn and $res = join ’’,<FF> and close FF or die;
    my $sm = $res=~/,streaming:(\[[^\]]*\]),/ ? $1 : die;
    $sm=~s/(\w+):/"$1":/g;
    my $title = $res=~m{<title>([^<:]+)} ? $1 : die;
    $title=~s/\x22/\x27/g; #double to single quote
    return {q=>$q, title=>$title, rel=>JSON::XS->new->decode($sm)};
}
my %res = ();
sub needrel{
    map{ $res{$_} ||= get($_) } map{ $_ ? "$$_{e}%3A$$_{s}" : () }
    map{ @{$$_{rel}||die}[1..$relcnt] } @_
}

needrel({rel=>[0,{qw[e NYSE s IBM]}]});
needrel(values %res) for 1..$depth;
#print scalar keys %res;
#print JSON::XS->new->encode(\%res);
#print sort map{"$$_{q} $$_{title}\n"} values %res;
(values %res) == uniq(map{$$_{title}} values %res) or die;
print sort map{
    my $t=$$_{title}; map{qq["$t","$$_{title}"\n]} needrel($_)
} values %res;
```