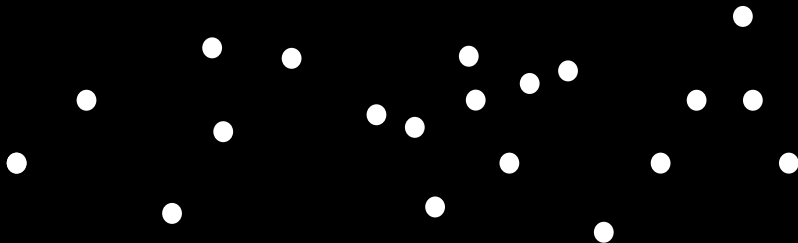


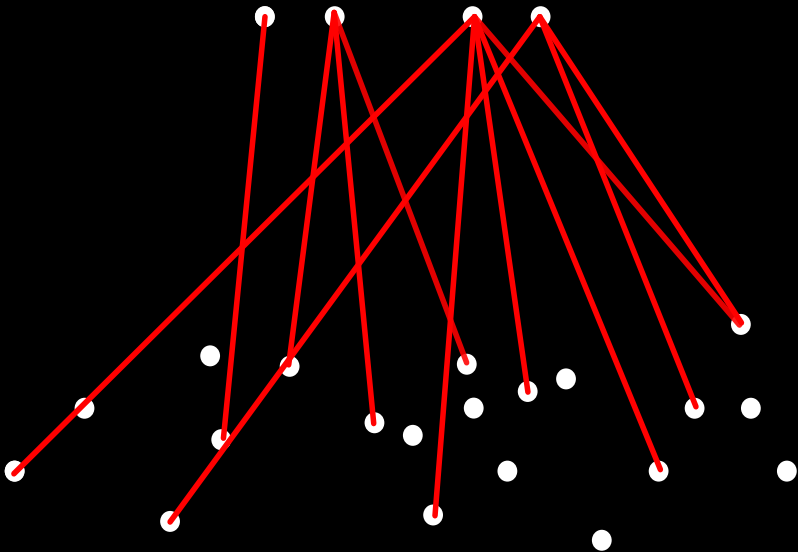
Iterated Compression

Try, try, till you succeed – or fail.

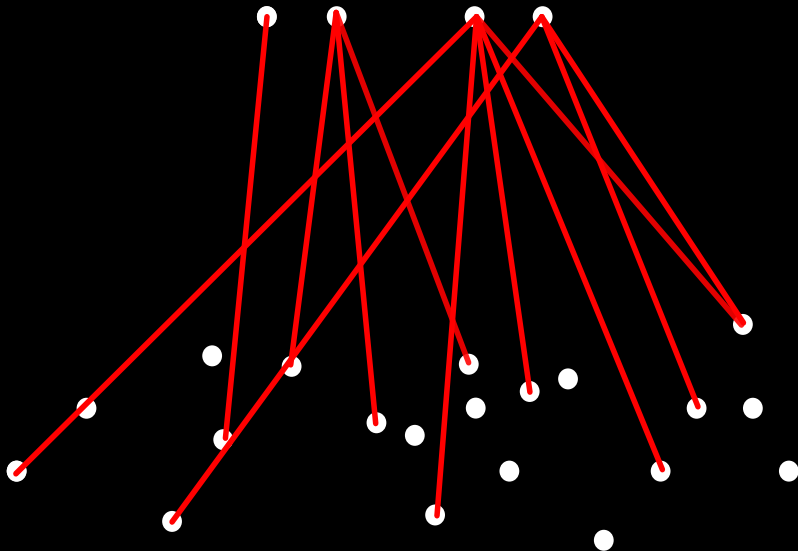


Schematic Representation of Data

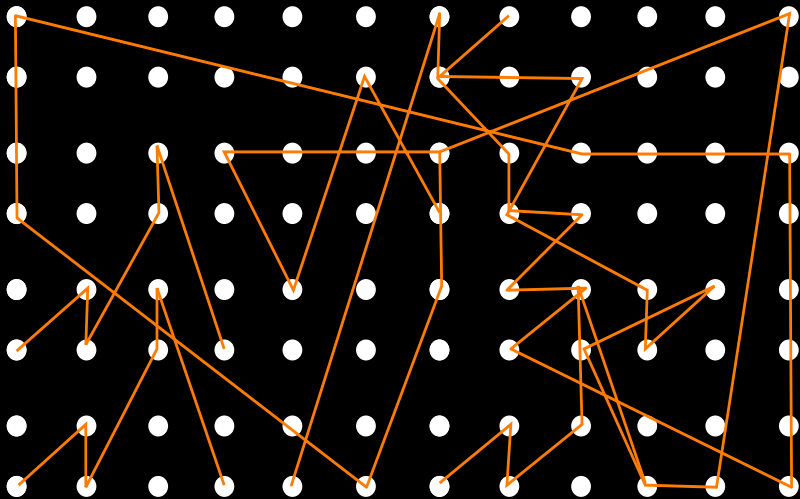




Schematic Representation of Conflicts



Unfortunately, the picture could be uglier.



Let's look at some numbers.

For a respectable hypothesis,
at most 0.01% of data points may be
declared “bad”.

For a respectable hypothesis,
10 out of 1000 of data points may be
declared “bad”.

Try all subsets of 10 points as possible candidates for elimination?

Even if you handled ten thousand subsets in one second, you would have to spend

$$\binom{1000}{10} / 10000 \approx 1000^6$$

seconds.

THAT
is the age of the universe.

THAT
is the age of the universe.
Ten times over.

That's what a stupid

$$n^{n/100}$$

algorithm does to you.

What about an algorithm with

$$2^{n/100} \cdot n^2$$

operations?

What about an algorithm with

$$2^{10} \cdot 1000^2$$

operations?

Assuming ten thousand operations to
a second, this is:

$$1024 * 100$$

A couple of days.

Use k to denote the size of the solution we seek.

Use k to denote the size of the
solution we seek.

Typically, k is a small fraction of the size of the input.

n^k

$C^k n^d$

You want to find a vertex cover of size k in a graph that has n vertices.

You want to find a vertex cover of size k in a graph that has n vertices.

Your friend gives you a vertex cover of size $(k + 1)$.

You want to find a vertex cover of size k in a graph that has n vertices.

Your friend gives you a vertex cover of size $(k + 1)$.

Something is better than nothing?

You both show it to your professor, who spends some time on a graph and comes up with VC of size k .

Story Over?

No, because your professor doesn't give you the solution! :(

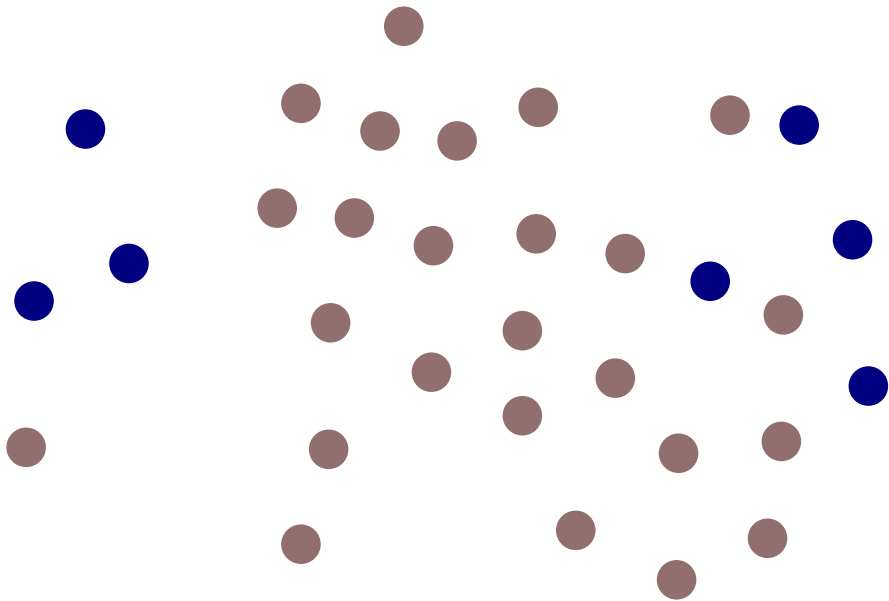
No, because your professor doesn't give you the solution! :(

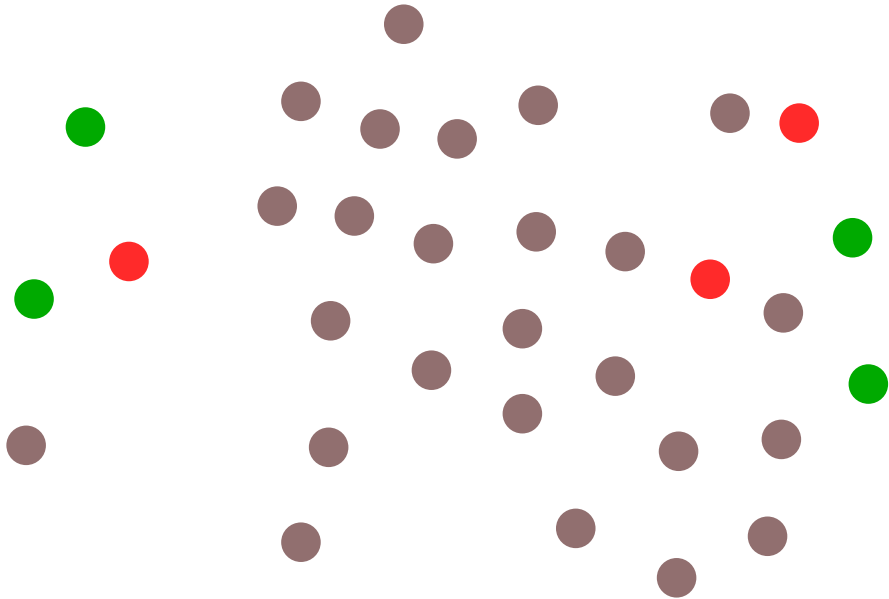
He is a nice person, though, so he tells your solution was pretty close.

No, because your professor doesn't give you the solution! :(

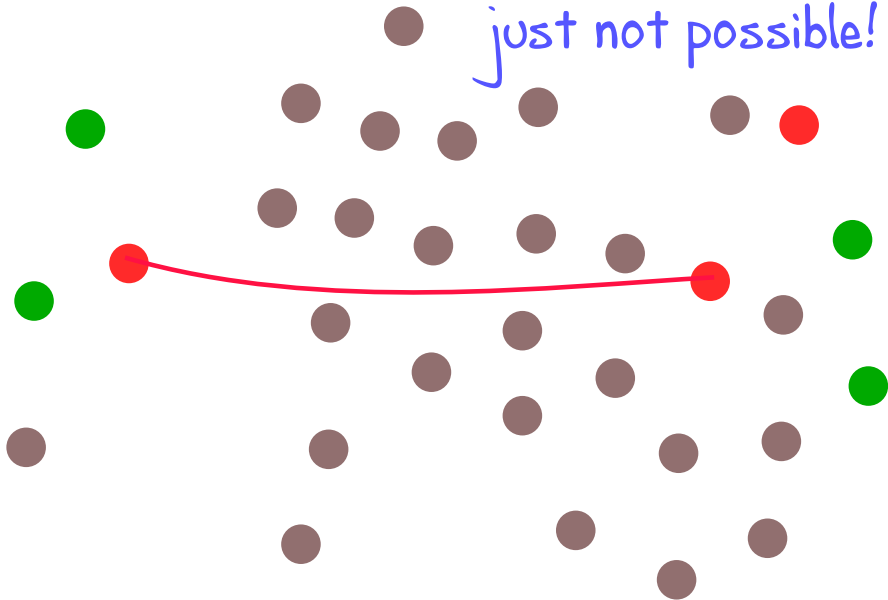
He is a nice person, though, so he tells your solution was pretty close.

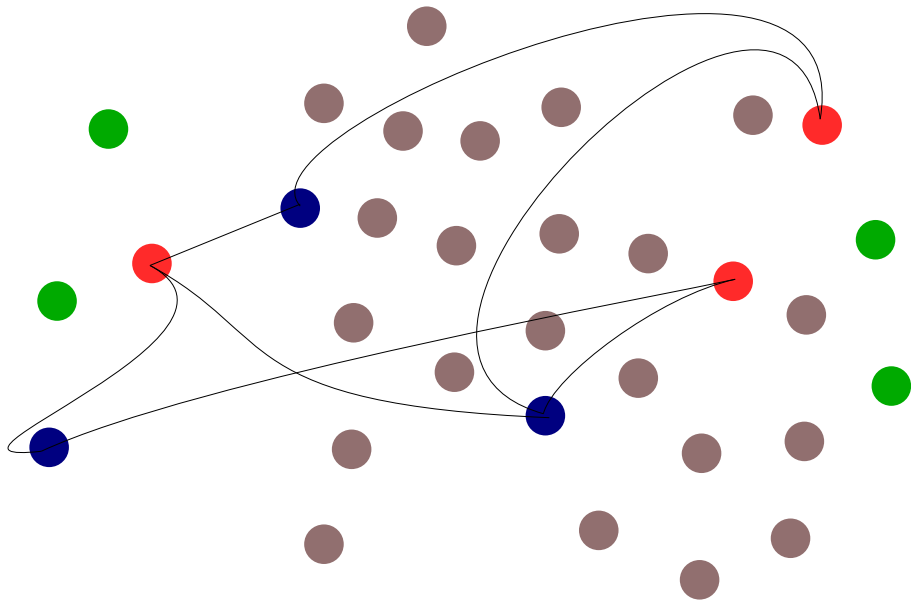
And he points out which of the vertices in your solution also feature in *his* solution.

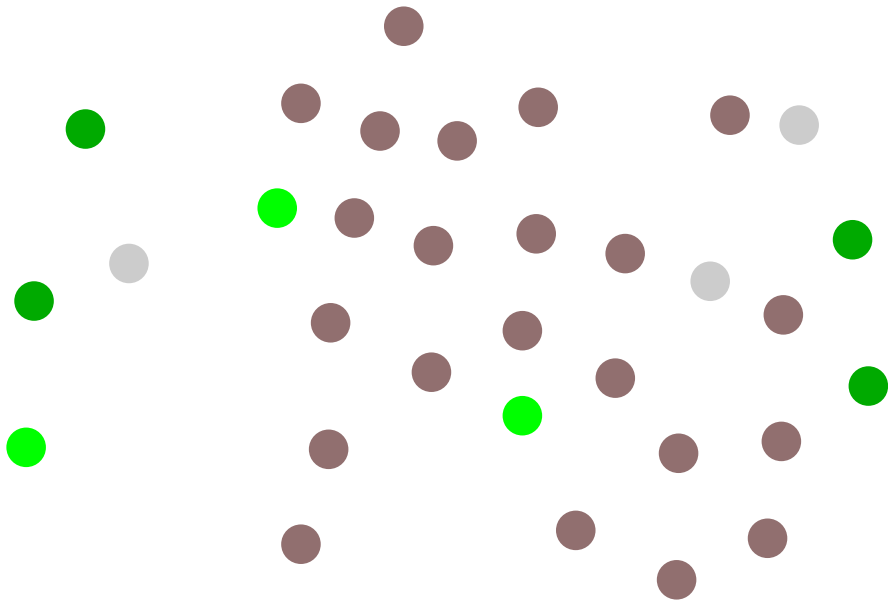




just not possible!







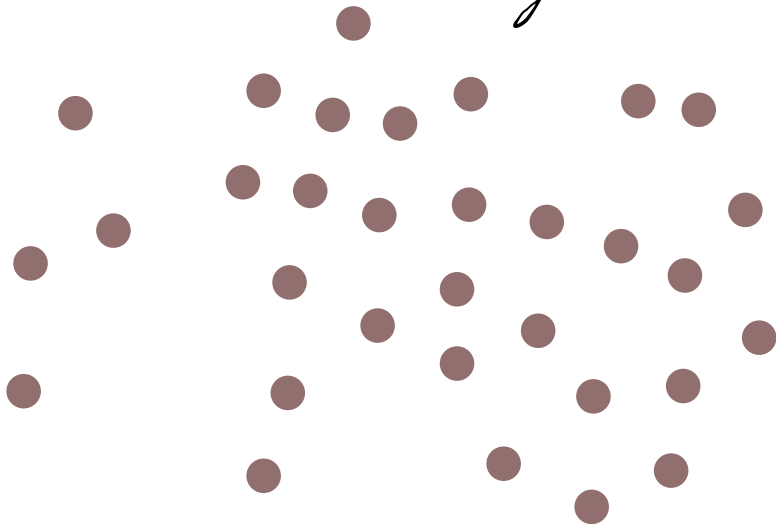
Story Over :)

A More General Story

Part 1: What you do when you don't have a nice professor.



your solution





partly right, partly wrong
with respect to some mystery solution.

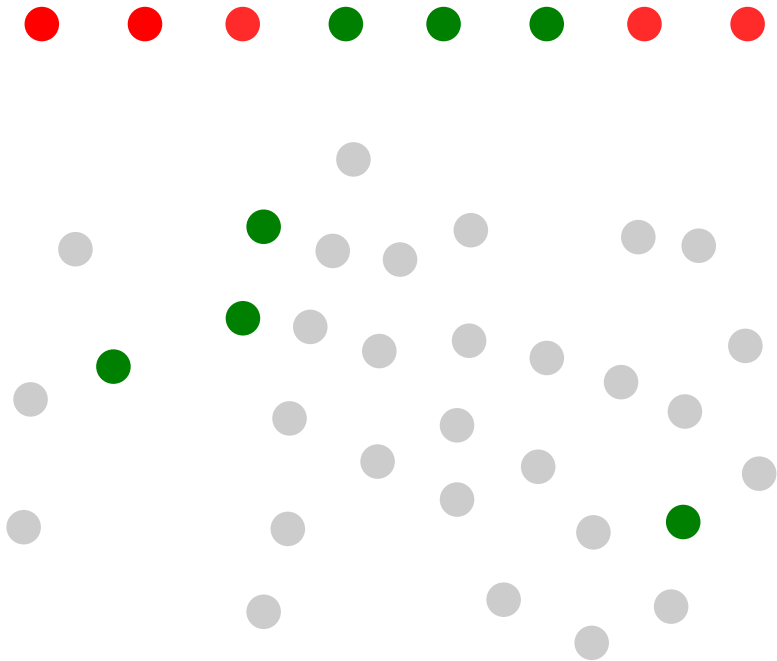


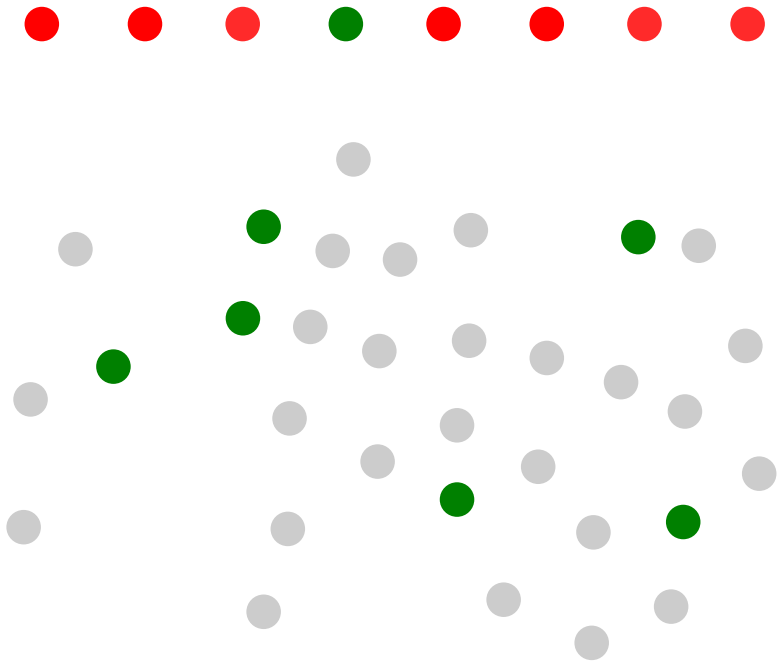
We don't know where we are right,
and where we are wrong.

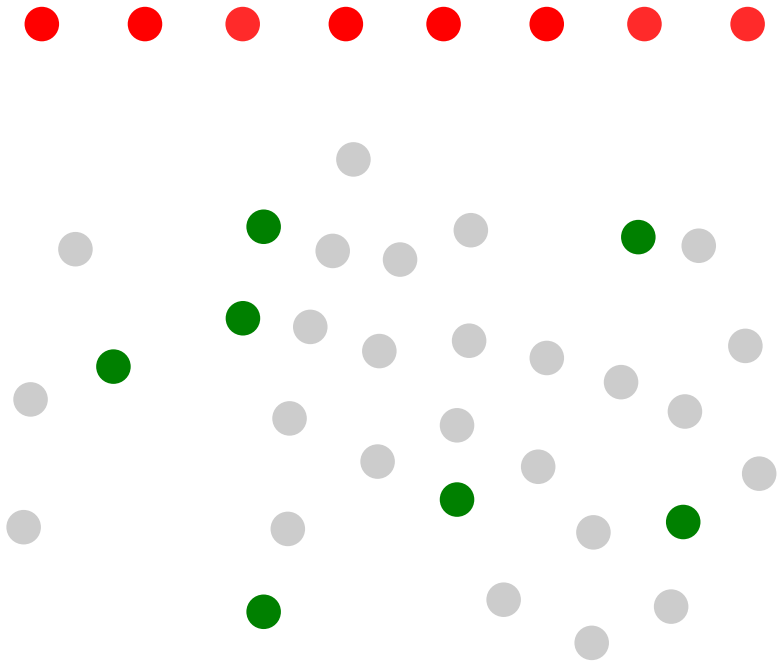
We don't know where we are right,
and where we are wrong.

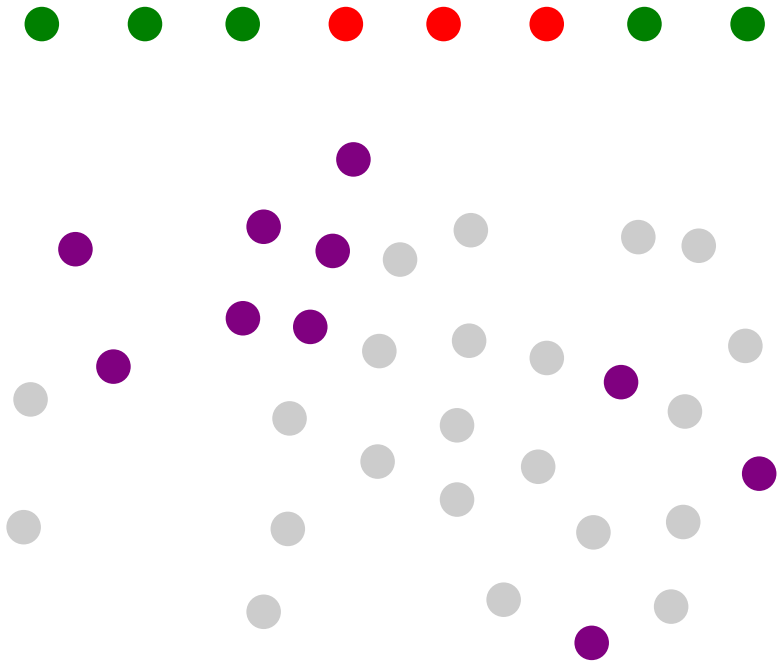
So, we go over **all possible**
right/wrong possibilities, and try
them out.

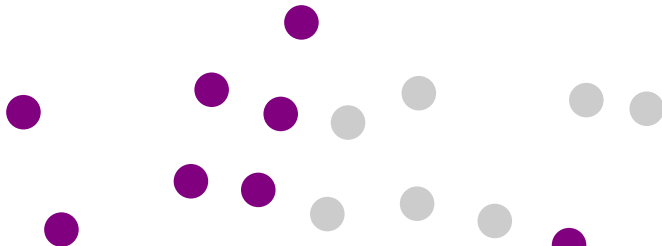
If mystery solution exists, we will find
it one of our runs.



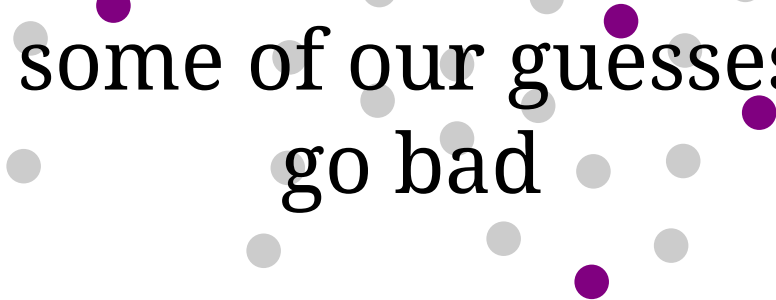








some of our guesses
go bad



But that's okay, we just move on
from them.

But that's okay, we just move on
from them.

So what's the price we pay for the
missing professor?

If your friend gives you a solution of size k , then how long will it take you to go over all right/wrong possibilities?

Coming up with a smaller solution
from a larger one is called
Compression.

Compression usually involves coming up with ideas for figuring out what to do with each of those possibilities.

Compression usually involves coming up with ideas for figuring out what to do with each of those possibilities.

But we have learnt what to do without the helpful professor.

Compression usually involves coming up with ideas for figuring out what to do with each of those possibilities.

But we have learnt what to do without the helpful professor.

Remember that it cost us 2^k .

A More General Story

Part 2: What you do when you don't
have a friend.

The compression worked on a solution that was really close (as far as *size* goes) to the one you were looking for.

The compression worked on a solution that was really close (as far as *size* goes) to the one you were looking for.

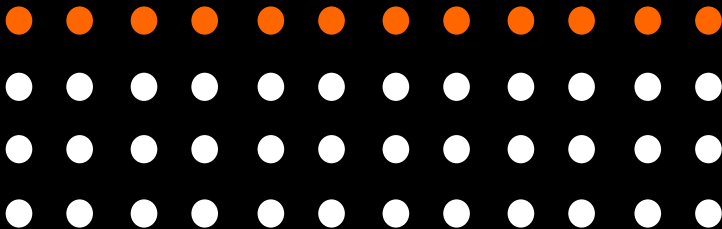
But suppose your kind friend is on vacation, and you think a solution of $k + 1$ is about as difficult as finding a solution of size k .

The compression worked on a solution that was really close (as far as *size* goes) to the one you were looking for.

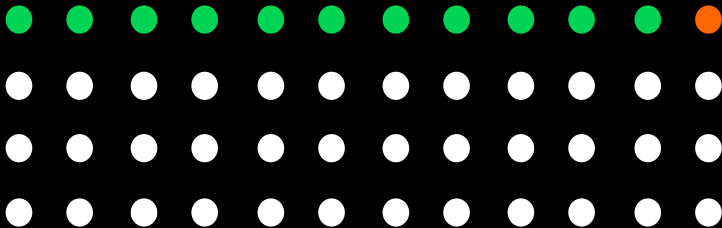
But suppose your kind friend is on vacation, and you think a solution of $k + 1$ is about as difficult as finding a solution of size k .

Now what?

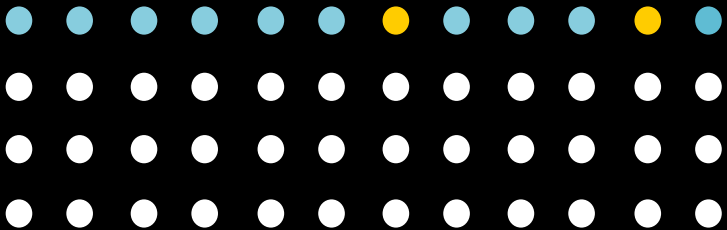
We'll do something really, really
clever.



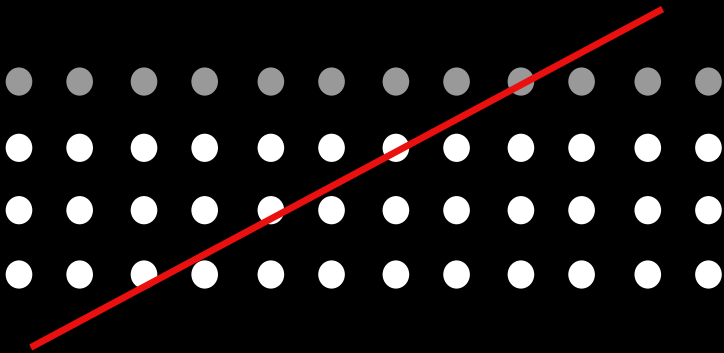
start with any $k+2$ vertices



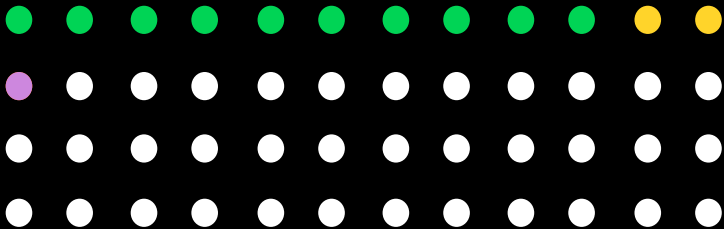
start with any $k+2$ vertices
any $k+1$ vertices form a vertex cover



compress it!

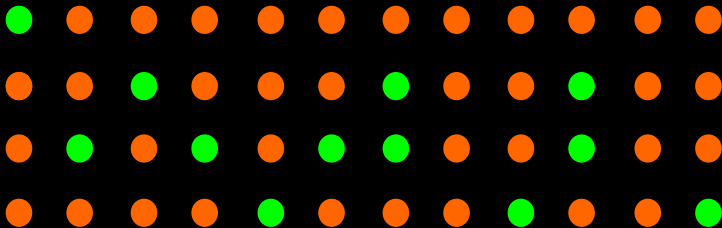


if compression fails, then there is
no vertex cover for the whole graph (why?)

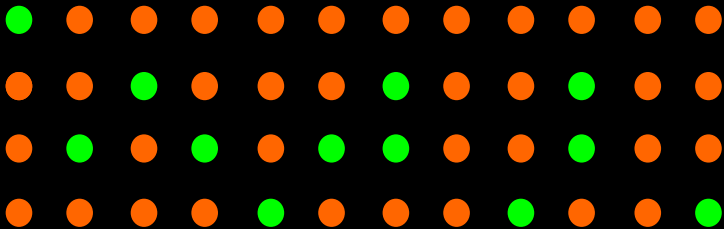


else, continue by expanding your subgraph
with one more vertex,
and add that to the k -sized solution
(to get a $k+1$ sized solution).

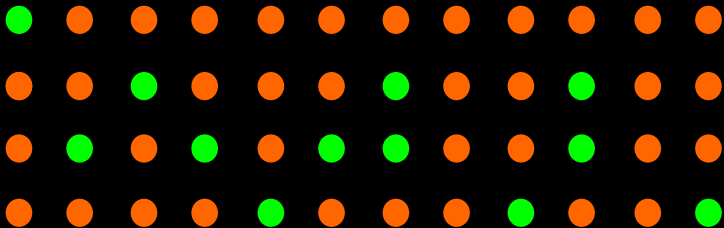
Slowly, but surely, we'll cover the whole graph.



If you don't say no,
you will say yes.



If you don't say no,
you will say yes.
And conversely!



If you don't say no,
you will say yes.
And conversely!
So - at the end of it - you know.

We'll either stop en route and say
NO or else when we stop, we have a
solution.

What time does it take to do this?

$n \cdot$ (Time taken by Compression
Routine)

The process of repeating the
Compression Routine many times
over is called Iteration.

So your friend's vacation cost you a polynomial factor.

So your friend's vacation cost you a polynomial factor.

But now you can do it all on your own!

So your friend's vacation cost you a polynomial factor.

But now you can do it all on your own!

This will be our mantra: *iterate*, *compress*, *iterate*, *compress*, *iterate*... till you say no, or till you got them all.

End of The General Story :)

The Particular Stories

FeedBack Vertex Set

Find a subset of at most k vertices whose removal destroys all cycles.

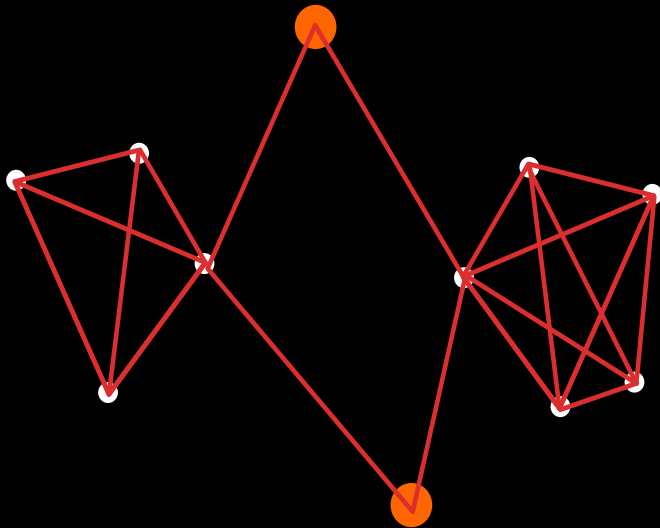
Applications in operating systems (removal of deadlocks), genome assembly, and VLSI chip design.

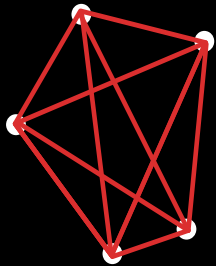
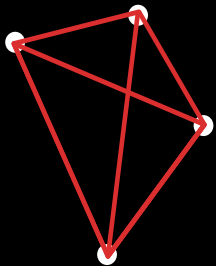
Odd Cycle Transversal

Find a subset of at most k vertices whose removal destroys all cycles of odd length.

Cluster Vertex Deletion

Find a subset of at most k vertices whose removal makes the graph a collection of clusters.





Terms We Use

NP-complete

NP-complete

NP-complete

In the real world, you don't tell your
boss that your problem is
NP-complete.

Parameterization

Fixed-Parameter Tractability

Fix a parameter and the problem becomes tractable!

$C^k n^d$

That will be all – Philip has more.

That will be all – Philip has more.

He'll take all the questions too.