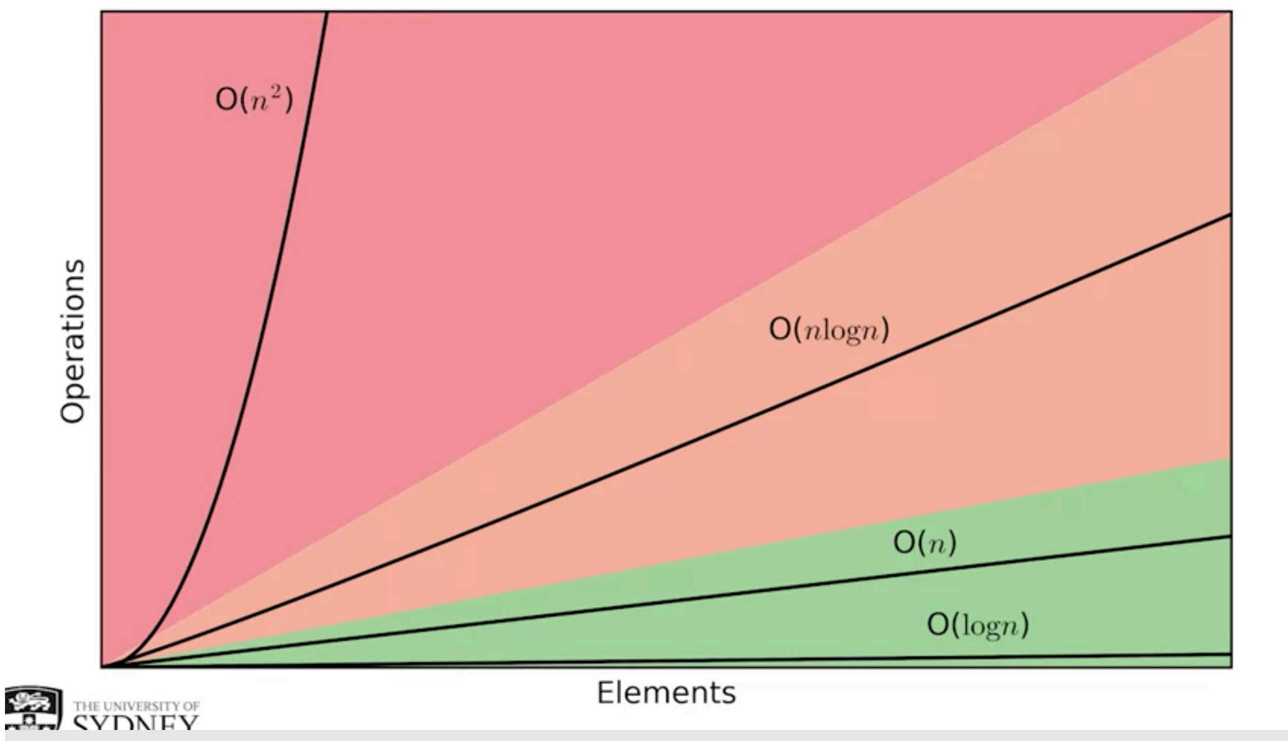
Time Complexity Comparison and Conclusion

Our Naive Cross matching algorithm, when we increase the number radio galaxies in each catalogue by a factor of 10 then the running time is observed to increased by a factor of 100 (10*10)

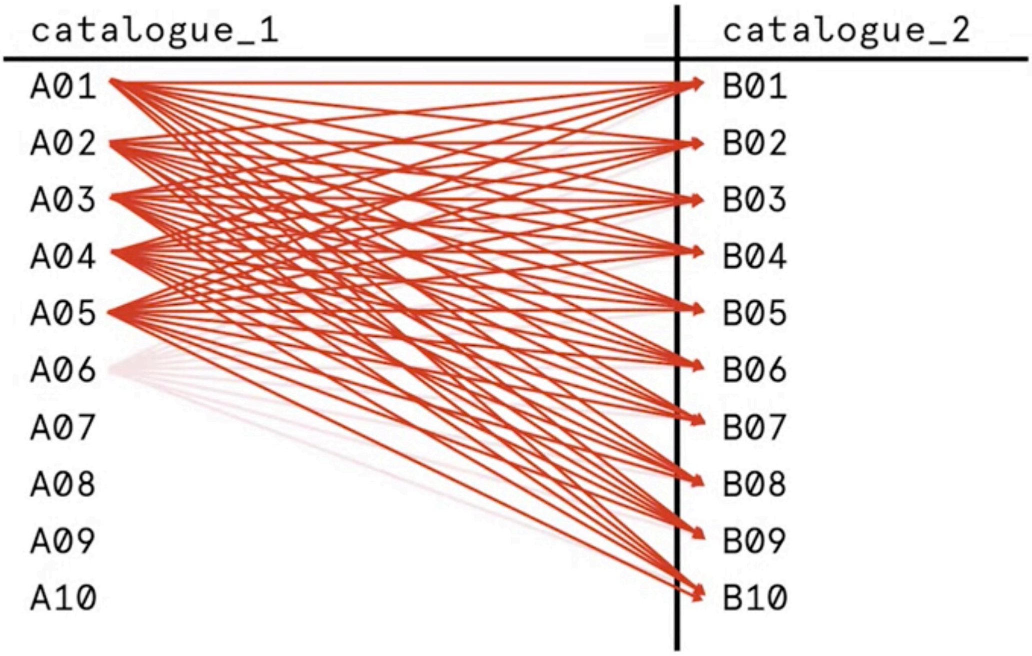
if we deliberately increase the size of both catalogues by same amount but generally we could say that our algorithms time complexity in bigO notation is scaling by n times m
where
Catalogue 1 : n galaxies
Catalogue 2 : m galaxies

time complexity is a(nm)+b(n)+c(n) +d in Big-O - O(nm)



an algorithm is to scale very poorly according to the above plot once we’ve worked with how our algorithms work based on the complexity, we need to think of ways to improve it.The chances are that if you've done a really naive implementation, like we have, there are plenty of things you can do to speed it up.

In the case of our cross-matching, the reason our code is scaling as N by M, is that every single galaxy in the first catalog is being compared against every single galaxy in the second catalog. And each time a great circle distance is calculated.



Calculating this distance is the expensive step. So is there a way we could reduce how many times we need to do it? I'm sure you've got plenty of ideas, but as an example, let's just try one simple one

Many ways to speed up a program:

- Use a faster programming language;
- Incorporate better data structures;
- Include specialised routines;
- Streamline your code;
- Improve time complexity of core algorithm.

THE UNIVERSITY OF SYDNEY

Improving our Algorithm

A much faster algorithm

The astropy module has a cross matching algorithm included which is extremely faster

```python
from astropy.coordinates import SkyCoord
from astropy import units as u
c = SkyCoord(ra=ra1*u.degree, dec=dec1*u.degree)
catalogue = SkyCoord(ra=ra2*u.degree, dec=dec2*u.degree
idx, d2d, d3d = c.match_to_catalog_sky(catalogue)
```

It is used to calculate angular distances and cross match two catalogues

So how fast this will run ?

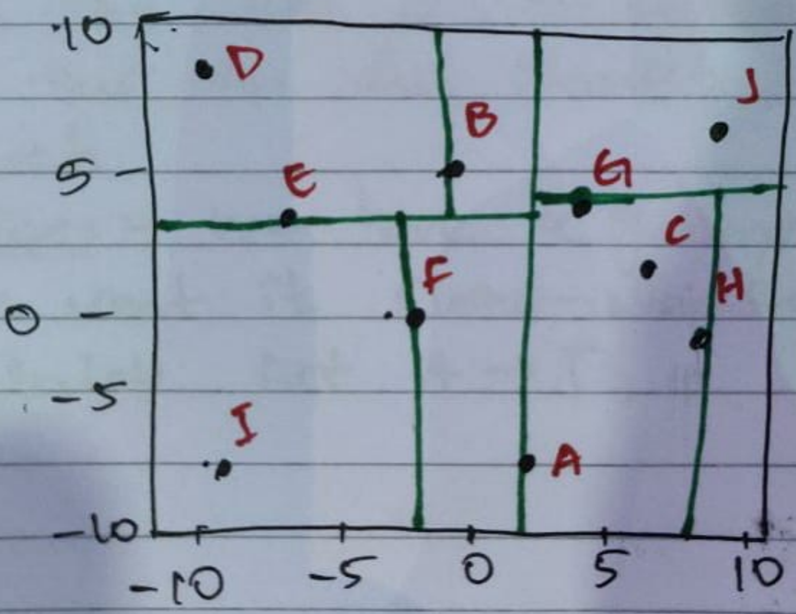
Considering our initial algorithm and the slightly improved one (in the video) the first one took 2secs for 1000x1000 data points in each catalogue (i.e 1000 in one catalogue 1000 in other)
that means for a typical 1million x 1million data-points in catalogues it will take 2 s i.e 24 days of continuous computation.
The slightly improved one on the other hand takes 1 second for 1000x1000 and takes half a million for million x million data which is 12 days

the Astropy Crossmatching however takes **25secs!**

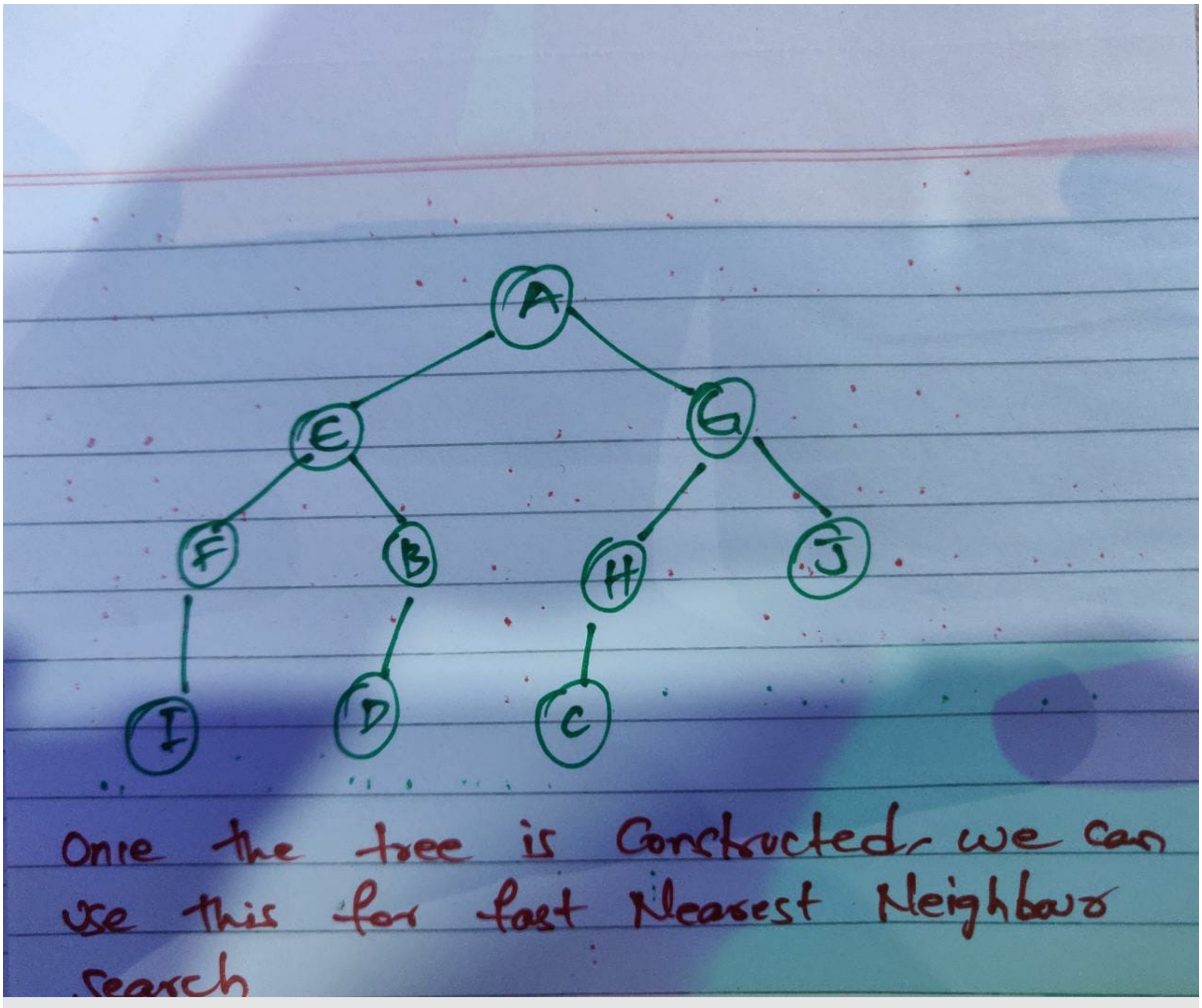
K-d Tree (k- dimension Tree)

24 days → 12 days → 25 seconds!
How did Astropy Achieve this?
. . .

Data Structure called k-d tree or k-dimensional tree is a way repre-senting points in Space in a recursive Structure (k = No of Dimensions)
In our Case its 2D, Right ascension and Declination.



first split is at median of datapoints (A) root node, then Recursively splitting y or x Alternatively. In the left plane and Same on the Right



We not only did change our implementation but also happen to improve our time complexity

if we need to cross match large dataset, we could use databases to improve the efficiency and reduce the cpu cycles

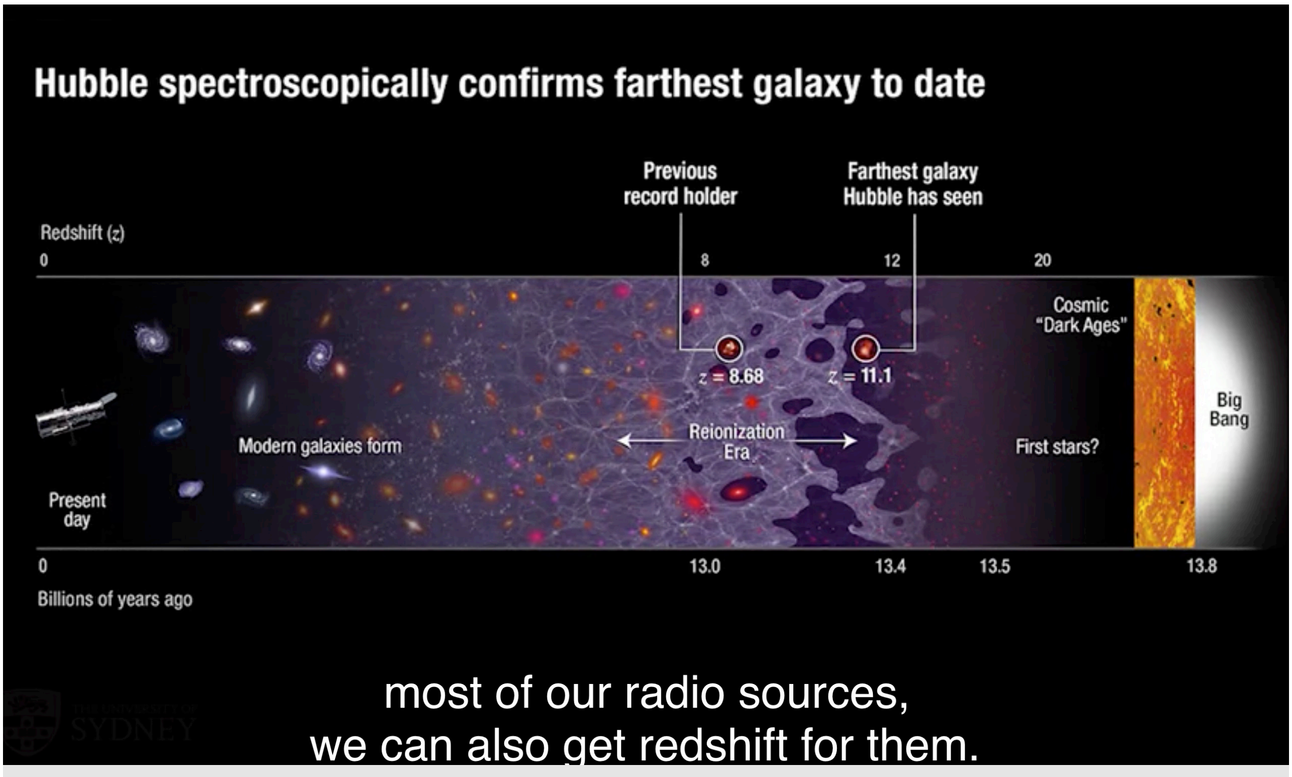
Another important issue is how you can evaluate whether your matches
are just chance coincidences.To answer this question, you may need a combination of astronomy problem solving and computational thinking.

1. For example, if you can measure redshift for your objects, you can establish that they're at the same distance. And then it's much more likely they're physically associated.
2. You could also do a statistical analysis to calculate the likelihood of a chance
3. coincidence given the spatial density of objects in the two surveys.

So now we've done our cross-matching successfully. What do the results actually mean?

~ Now we know that all of our radio sources have optical counterparts, which means we can classify them into two different categories.

1. Most of our radio galaxies are associated with quasars. Where we're looking towards the central black hole and can see the very energetic accretion disk. The radiation from the accretion disk is so bright that it outshines all of the stars in the galaxy. And therefore, looks just like a bright star, hence the name, **quasi-stellar object, or quasar.**
2. The rest of our radio galaxies sit inside normal galaxies, where we can see a cloud of many stars grouped together. This could mean that the supermassive black hole has stopped accreting material. And the radio jets are remnants of past activity.



The redshift for the galaxies range from 0.02 to 0.5,
whereas the redshift for the quasars ranges from 0.2 to 3.
This tells us most of the galaxies are in the relatively local universe,
whereas most of the quasars are typically much further away.

By matching the optical and radio catalogs,
we've been able to see the different types of galaxies that can host supermassive
black holes and measure the distance to them.

This is just one example of the additional science that can be done
by combining information from different wavelengths.
And illustrates why most of modern astronomy takes a multi-wavelength
approach.