Hashing,

Also know as a hash map

Explain what a hash table is- use to store key value pairs and are not orderd, its fast to add, store and remove. It must be human readable and it helps the pc read everything

Hash function- they are used all around the world

Define it-

Takes data of a arbitrary size and returns a fixed data size

What makes it good –

Must be fast

It doesn’t cluster

Deterministic(same output in connection to the input)

Simple hash-

“a”.charCodeAt(0)// will give the characher value

The smaller the string/array the faster the hash.

Seprate changing is joining multiplue value and putting it in a nested data structure [ [ ] ]

Linear probing- putting the value into the next available slot

How collusions work

Seprate changing

Is actually built into most programming languages

Its use because it is affeciatnt

It takes about 0,1ms to work and is not tide to the elements stored in the table

The way it works

It takes a key input and runs it through a hash function and it maps strings to numbers and correspond to indexes in a array.

It needs to be consistent so the same word gives the same number

A collision is when same input(name) they are turned into the same index number.

Big O Notation

Simplified analysis of an algotithms efficiency

Constant time

X = 5 +(15\*20);

It computes x it is a big O of 1 O(1)

How do we get the following

X=5+(15\*);

Y=15-2;

Print x = y;

We take the time it computes to get each answer O(1) then multilply it but we are ignoring constants so it will still be O(1)

Linear time-

For x in range (0, n):

Print x;

So in this example the answer would be the big O(n) because the question is to go through everything until n…..

More complicated one next-

Y= 5 + (15\*20);

For x in range (0, n):

Print x;

So the first line we know is O(1) and the second is O(n) but because we drop low order terms, so basicicly the for loop dominates the run time so the answer will then be O(N)

Quadratic time

For x in range (0, n):

For y in range (0, n):

Print x \* y;

So this one will be 2 for loops and it will be O(N squared)

Need for it-

How to know what code is the best //as they spoke about yesterday saying the shortest way is normally the best way to do something

Performance matters in the large scale

What is better?

Faster, less memory intensive or more readable?

It matters from situation to situation.

Rather than counting seconds we can count the operations the pc has to perform

Big o notation is basically the time it takes how long the function runs in connection to the size of the algorithm

Logartithms

Big O expressions that aren’t common

Invers of exponents

You see it with –

Certain searching algorithms have logarithmiv time complexity

Efficient sorting involve it

Recursion sometimes involve logarithmic space complexity

Recap-

To analyse the performance of an algorithm, we use Big O Notation

Big O Notation can give us a high level understanding of the time or space complexity of an algorithm.

Big O Notation doesn’t care about precision, only about general trends.

The time or space complexity (as measured by Big O) depends only on the algorithm.

Big O Notation is everywhere

A good way to think about it is a rector scale, it shows how “bad” the code is that is running