OCR Computer Science AS Level

2.3.1 Searching Algorithms

Intermediate Notes

Specification:

- Standard algorithms
  - Binary search
  - Linear search

Searching Algorithms

Searching algorithms are used to find a specified element within a data structure. For example, a searching algorithm could be used to find the name “Alan Turing” in an array of names.

Numerous different searching algorithms exist, each of which is suited to a particular data structure of format of data.

Binary Search

The binary search algorithm can only be applied on sorted data and works by finding the middle element in a list of data before deciding which side of the data the desired element is to be found in. The unwanted half of the data is then discarded and the process repeated until the desired element is found or until it is known that the desired element doesn’t exist in the data.

Pseudocode for binary search is shown below.

A = Array of data
x = Desired element

```
low = 0
high = A.length -1
while low <= high:
     mid = (low + high) / 2
     if A[mid] == x:
          return mid
     else if A[mid] > x:
          high = mid -1
     else:
          low = mid + 1
     endif
endwhile
return “Not found in data”
```

With each iteration of binary search, half of the input data is discarded, making the algorithm very efficient.

Example 1

Find the location of “Dylan” in the data below.

0	1	2	3	4	5	6
Bob	Charlie	Dylan	Ellie	Franz	Gabbie	Hugo

To start with, our values for high and low and 6 and 0 respectively.
The first step is to find the middle position. In this case, it’s `(0 + 6) / 2 =` **3**.
Next we inspect the data in position 3: Ellie. This is higher than the desired data and so we discard elements 3-6, setting high as 2.

0	1	2	3	4	5	6
Bob	Charlie	Dylan	Ellie	Franz	Gabbie	Hugo

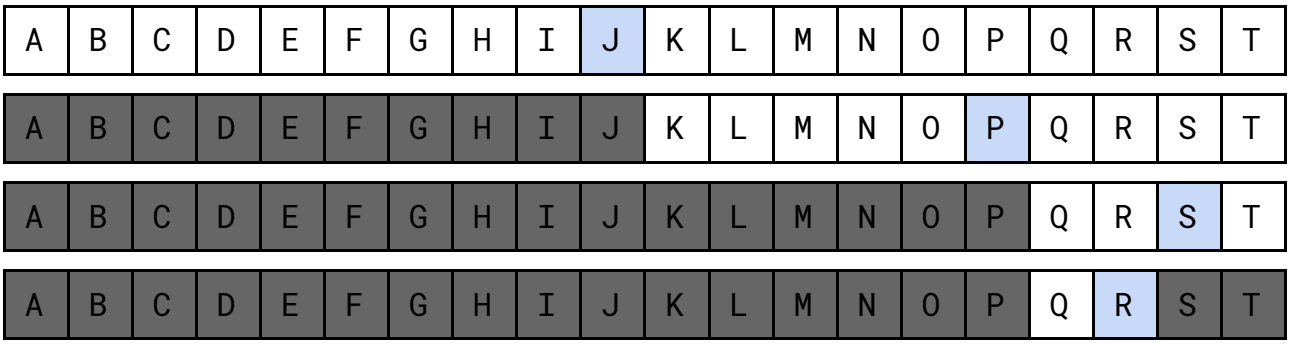
Again, we calculate the value of the middle position. This time it’s `(0 + 2) / 2 =` **1**.
Inspecting the data at this position we find Charlie, which is lower than the data we’re looking for. We now set low to 2, discarding items 0 to 1.

0	1	2	3	4	5	6
Bob	Charlie	Dylan	Ellie	Franz	Gabbie	Hugo

Calculating the middle position, we have `(2 + 2) / 2 =` **2**.
Inspecting the item in position 2, we find Dylan. This is our desired data and so we return the position 2.

Example 2

Look at how the algorithm finds the letter R in the first 20 characters of the alphabet. It's clear from this example that the algorithm halves the remaining data to be searched with each iteration, gradually reducing the size of the problem to be solved.



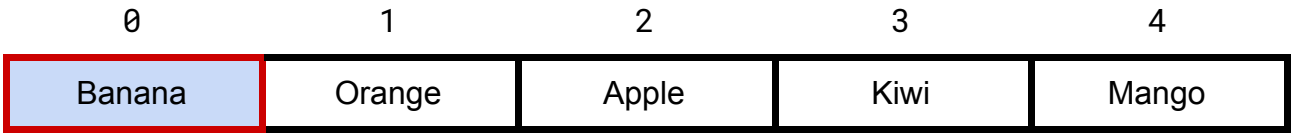
Linear Search

Linear search is the most basic searching algorithm. You can think of it as going along a bookshelf one by one until you come across the book you're looking for. Sometimes the algorithm gets lucky and finds the desired element almost immediately, while in other situations, the algorithm is incredibly inefficient.

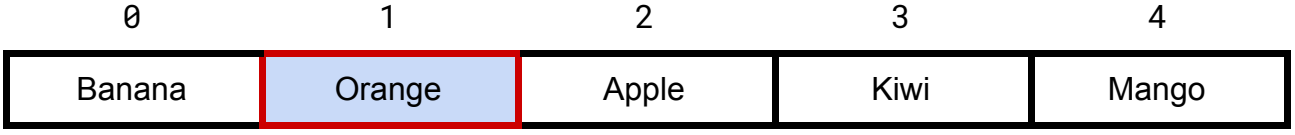
There's a lot of pot luck involved with linear search, but it's incredibly easy to implement. Unlike binary search, linear search doesn't require the data to be sorted.

Example 1

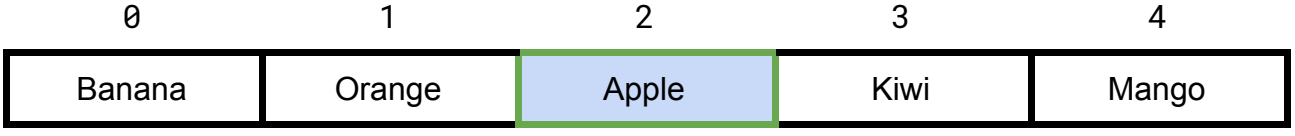
Find the position of Apple in the data.



First we inspect position 0, and find Banana. Not the element we're after.



Next we inspect position 1, finding Orange. Again, not what we're looking for.



Next we look at position 2 and find Apple, this is the data we're looking for and so the algorithm returns 2 and terminates.

<u>Example 2</u>

Look at how this algorithm finds the letter R in the first 20 characters of the alphabet and compare it to binary search above. It's clear from this example that the algorithm is much less efficient than binary search.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T