

Binary Search Examples

Suppose we wish to search the sequence/list/array below for two keys. We shall search using the binary search method, using Algorithm 5.3.2 on pages 250 and 251 of the text. Note that in real world applications, the data stored would not be as simple as that below. Usually the data consists of some sort of key term, such as an ID number, and additional data, such as name and address, etc. However to illustrate the searching algorithm we shall use data consisting of a single character.

A	G	M	P	Q	U
---	---	---	---	---	---

Example 1

Search the above list for the key K .

Call 1

1	2	3	4	5	6
A	G	M	P	Q	U
i					j

Note that i and j are 1 and n respectively at the beginning. The part of the list presently being searched is in boldface type. At line 4, k is set to $\lfloor (1+6)/2 \rfloor = \lfloor 3.5 \rfloor = 3$.

Key is now compared to $s_k=M$, at line 5. They are not equal, so we drop through to line 7. Key is now compared to s_k again, to see if key is less than s_k . Key (K) is less than s_k (M) so we drop j to $k-1$. Now make a recursive call to *BinarySearch*.

1	2	3	4	5	6
A	G	M	P	Q	U
i		k			j

Call 2

1	2	3	4	5	6
A	G	M	P	Q	U
i	j	k			

At the start of the second call, i is 1 and j is 2. So k becomes $\lfloor (1+2)/3 \rfloor = 1$.

1	2	3	4	5	6
A	G	M	P	Q	U
i, k	j				

So we compare s_i to key, and find they are not equal. Then we check again to see if $key < s_k$, and it is not. Hence we increase i to $k+1=2$, as follows.

1	2	3	4	5	6
A	G	M	P	Q	U
<i>i,k</i>	<i>j</i>				



Another call is made, call 3, with $i=j=2$.

Call 3

1	2	3	4	5	6
A	G	M	P	Q	U
	<i>i,j</i>				

k is set to $\lfloor (2+2)/2 \rfloor = 2$. We compare key to $s_k = s_2$ and find they are not equal. Check to see if $key < s_k$ and it is not. Hence we increase i to $k+1$ and make a recursive call with the new i and j .

1	2	3	4	5	6
A	G	M	P	Q	U
	<i>i,j</i>				



Call 4

1	2	3	4	5	6
A	G	M	P	Q	U
	<i>j</i>	<i>i</i>			

This time the test $i > j$ returns true, so we return the value 0, meaning that key (K) was not found in the list. ♦



Example 2

Search the above list/sequence/array for the letter U .

Call 1

Note that at the beginning of the first call, i and j are 1 and n (again) and $key=U$.

1	2	3	4	5	6
A	G	M	P	Q	U
<i>i</i>					<i>j</i>

At line 4 we set k to $\lfloor (i+j)/2 \rfloor = \lfloor 7/2 \rfloor = 3$.

1	2	3	4	5	6
---	---	---	---	---	---

A	G	M	P	Q	U
<i>i</i>		<i>k</i>			<i>j</i>

Check to see if $s_k = s_3 = key$. It is not, so check to see if $key < s_k$. It is not, so move i up to $k+1$ and make a recursive call to *BinarySearch*.

1	2	3	4	5	6
A	G	M	P	Q	U
<i>i</i>		<i>k</i>			<i>j</i>

Call 2

1	2	3	4	5	6
A	G	M	P	Q	U
			<i>i</i>		<i>j</i>

Now $i=4, j=6$ so k is set to $\lfloor 10/2 \rfloor = 5$.

1	2	3	4	5	6
A	G	M	P	Q	U
			<i>i</i>	<i>k</i>	<i>j</i>

Check to see if s_k, s_5 , is equal to key . It is not, so check to see if $key < s_k$. It is not, so move i up to $k+1$ and make a recursive call to *BinarySearch* with $i=j=6$.

1	2	3	4	5	6
A	G	M	P	Q	U
			<i>i</i>	<i>k</i>	<i>j</i>

1	2	3	4	5	6
A	G	M	P	Q	U
					<i>i,j</i>

At line 4 we set k to $\lfloor (i+j)/2 \rfloor = \lfloor 12/2 \rfloor = 6$. The test to see if $s_i = key$ returns true, so we stop and return the index 6.

◆◆◆