

COMP 250

INTRODUCTION TO COMPUTER SCIENCE

Assignment Project Exam Help

<https://powcoder.com>

Week 3-3 : Recursion 2 (Binary Search)

Add WeChat powcoder

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- More recursive Algorithms
- Decimal to Binary conversion
- Power function
- Binary Search

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

RECALL: DECIMAL TO BINARY (ITERATIVE)

ALGORITHM

Constructing Base 2 Expansions

Assignment Project Exam Help

procedure *BinaryExpansion*(n)

$k := 0$ <https://powcoder.com>

While $n > 0$

$a_k := n \% 2$ Add WeChat powcoder

$n := n / 2$

$k := k + 1$

return $(a_{k-1}, \dots, a_1, a_0)$

Recall that a decimal number n requires approximately $\log_2 n$ bits for its binary representation.

DECIMAL TO BINARY (RECURSIVE)

ALGORITHM

Constructing Base 2 Expansions

procedure *BinaryExpansion*(n)

If $n > 0$:

BinaryExpansion($n/2$)

 print($n\%2$)

Also in this case, there are $\log_2 n$ recursive calls

POWER (x^n) – DEFINITION

- Definition of power

$$x^n = x \cdot x \cdots x$$

Assignment Project Exam Help
 n times

- Inductive definition:

<https://powcoder.com>

- Base clause:

Add WeChat powcoder

$$x^0 = 1$$

- Inductive clause:

$$x^n = x \cdot x^{n-1}$$

POWER (x^n) – ITERATIVE 1

Let x a positive integer and let n be a positive number.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
power(x, n) {  
    int result=1;  
    for(int i=1; i<=n; i++) {  
        result = result *x;  
    }  
    return result;  
}
```

POWER (x^n) – RECURSIVE

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
power(x, n) {  
    if(n==0) {  
        return 1;  
    } else {  
        return x*power(x,n-1);  
    }  
}
```

POWER() – CAN WE DO BETTER?

More interesting approach using recursion:

Assignment Project Exam Help

$x^{18} = x^9 * x^9$
<https://powcoder.com>

$x^9 = x^4 * x^4 * x$
Add WeChat: powcoder

$x^4 = x^2 * x^2$

$x^2 = x * x$

POWER (x^n) – RECURSIVE 2

```
power( x, n) {  
    if (n == 0)  
        return 1;  
    else if (n == 1)  
        return x;  
    else{  
        tmp = power(x, n/2);  
        if (n%2==0)  
            return tmp*tmp;        // one multiplication  
        else  
            return tmp*tmp*x;      // two multiplications  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

A SIMILAR IDEA CAN BE IMPLEMENTED ITERATIVELY

IDEA: Let's use the binary expansion of n , say $n = (a_{k-1}, \dots, a_1, a_0)_2$.

Assignment Project Exam Help

Note that:

$$x^n = x^{a_{k-1}2^{k-1} + \dots + a_12^1 + a_0} = x^{a_{k-1}2^{k-1}} \dots x^{a_12^1} \cdot x^{a_0}$$

<https://powcoder.com>

This shows how to compute x^n . We only need to compute the values of $x, x^2, (x^2)^2 = x^4, \dots, x^{2^k}$. Once we have these terms we multiply the terms x^{2^j} , where $a_j = 1$.

Add WeChat powcoder

POWER (x^n) – ITERATIVE 2

```
power(x, n) {  
    result = 1;  
    pow = x;  
    if(n%2 == 1)  
        result = x;  
    n = n/2;  
    while(n != 0)  
    {  
        pow = pow * pow;  
        if(n%2 == 1)  
            result = result * pow;  
        n = n/2;  
    }  
    return result;  
}
```

Let $n = (a_{k-1}, \dots, a_1, a_0)_2$

<https://powcoder.com>

$\log_2(n) - 1$ iterations
// 1 multiplication

// 1 multiplication

POWER (x^n) – ITERATIVE 2

```
power(x, n) {  
    result = 1;  
    pow = x;  
    if(n%2 == 1)  
        result = x;  
    n = n/2;  
    while(n != 0)  
    {  
        pow = pow * pow;  
        if(n%2 == 1)  
            result = result * pow;  
        n = n/2;  
    }  
    return result;  
}
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

Let $n = (a_{k-1}, \dots, a_1, a_0)_2$
// $\log_2(n) - 1$ iterations
// 1 multiplication
// 1 multiplication

POWER (x^n) – ITERATIVE 2

```
power(x, n) {  
    result = 1;  
    pow = x;  
    if(n%2 == 1)  
        result = x;  
    n = n/2;  
    while(n != 0)  
    {  
        pow = pow * pow;  
        if(n%2 == 1)  
            result = result * pow;  
        n = n/2;  
    }  
    return result;  
}
```

Let $n = (a_k, \dots, a_1, a_0)_2$

<https://powcoder.com>

$\log_2(n) - 1$ iterations
// 1 multiplication

// 1 multiplication

EXAMPLE: x^{243}

$$n = (243)_{10} = (11110011)_2$$

Assignment Project Exam Help

<https://powcoder.com>

Q: How many multiplications do we need?

Add WeChat powcoder

EXAMPLE: x^{243}

$$n = (243)_{10} = (11110011)_2$$

Assignment Project Exam Help

<https://powcoder.com>

Q: How many multiplications do we need?

Add WeChat powcoder

A: Recursive method: $5*2 + 2*1 = 12$.

Iterative method: $7 + 5 = 12$

The highest order bit in the recursive method is the base case, and doesn't require a multiplication.

The lowest order bit in the iterative method does not require multiplication.

EXAMPLE: x^{243}

$$n = (243)_{10} = (11110011)_2$$

Assignment Project Exam Help

<https://powcoder.com>

Q: How many multiplications do we need?

Add WeChat powcoder

A: $O(\log n)$

OBSERVATIONS

The second approach we looked at uses fewer multiplications than the first one, and thus the second approach *seems faster*.

Assignment Project Exam Help

Q: Is this indeed the case? <https://powcoder.com>

Add WeChat powcoder

A: No. Why not ?

OBSERVATIONS

Hint: Let x be a positive integer with M digits.

Assignment Project Exam Help

■ x^2 has about ? digits.

<https://powcoder.com>

■ x^3 has about ? digits.

Add WeChat powcoder

■ :

■ x^n has about ? digits.

OBSERVATIONS

Hint: Let x be a positive integer with M digits.

- x^2 has about $2M$ digits.
- x^3 has about $3M$ digits.
- :
- x^n has about $n * M$ digits.

We cannot assume that multiplication takes 'constant' time.

Taking large powers gives very large numbers and multiplications becomes more expensive.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help

BINARY SEARCH

<https://powcoder.com>

Add WeChat powcoder

SEARCHING A LIST

- Goal: find a given element in a list.

Assignment Project Exam Help

- Solution: go through all the elements in the list and check whether the element is there (*linear search*).

<https://powcoder.com>
Add WeChat powcoder

- Could we do this any faster if the list was sorted to begin with?

Think of how you search for a term in an index. Do you start at the beginning and then scan through to the end? (No.)

- Exponential inequality 185
- Exponential running time 186, 661, 911
- Extended Church-Turing thesis 910
- Extensible library 101
- External path length 488, 839
- F**
- Factor an integer 919
- Factorial function 185
- Fail-fast design 107
- Fail-fast iterator 160, 171
- Farthest pair 210
- Fibonacci heap 628, 682
- Fibonacci numbers 57
- FIFO. *See* First-in first-out policy
- FIFO queue.
 See Queue data type
- File system 493
- Filter 60
 - blacklist 491
 - dedup 490
 - whitelist 8, 491
- Final access modifier 105–106
- Fingerprint search 774–778
- Finite state automaton.
 - See also* Maxflow problem
 - flow network 888
 - inflow and outflow 888
 - residual network 895
 - st-flow 888
 - st-flow network 888
 - value 888
- Floyd, R. W. 326
- Floyd's method 327
- for loop 16
- Ford-Fulkerson 891–893
 - analysis of 900
 - maximum-capacity path 901
 - shortest augmenting path 897
- Ford, L. 683
- Foreach loop 138
 - arrays 160
 - strings 160
- Forest
 - graph 520
 - spanning 520
- Forest-of-trees 225
- Formatted output 37
- Fortran language 217
- Fragile base class problem 112
- Frazer, W. 306
- Fredman, M. L. 628
- Function call stack 346, 415
- symbol tables 363
- type parameter 122, 134
- Genomics 492, 498
- Geometric data types 76–77
- Geometric sum 185
- getClass() method 101, 103
- Graph of a graph 519
- Global variable 113
- Gosper, R. W. 759
- Graph data type 522–527
- Graph isomorphism 561, 919
- Graph processing 514–693.
 - see also* Directed graph;
 - see also* Edge-weighted digraph; *See also* Edge-weighted graph; *See also* Undirected graph;
 - See also* Directed acyclic graph
- Bellman-Ford 668–681
- breadth-first search 538–541
- components 543–546
- critical-path method 664–666
- depth-first search 530–537
- Dijkstra's algorithm 652
- Kosaraju's algorithm 586–590
- Kruskal's algorithm 624–627
- longest paths 611, 612

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

BINARY SEARCH

- Inputs:
 - A sorted list.
 - The element we are looking for (the *key*)
- IDEA: First compare the key with the element in the middle of the list
 - If the key is less than the middle element, we only need to search the first half of the list, so we continue searching on this smaller list.
 - If the key is greater than the middle element, we only need to search the second half of the list, so we continue searching on this smaller list.
 - If the key equals the middle element, we have a match – return its index.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE

- Search for 25

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

-4	1	5	6	14	23	31	35	52	70
----	---	---	---	----	----	----	----	----	----

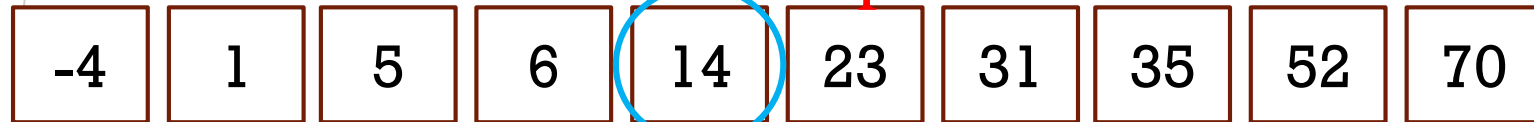
EXAMPLE

- Search for 25
- Look at the middle element and compare

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



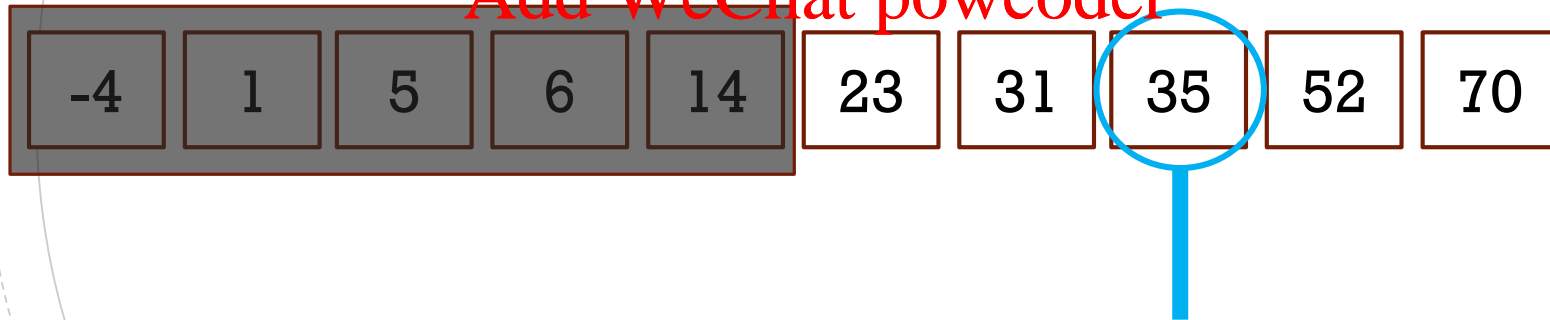
EXAMPLE

- Search for 25
- Look at the middle element and compare

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



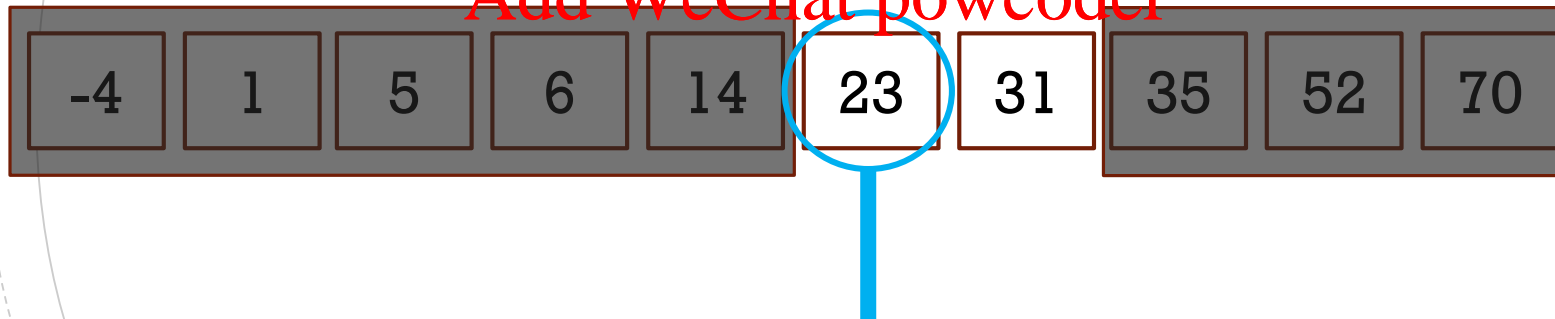
EXAMPLE

- Search for 25
- Look at the middle element and compare

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



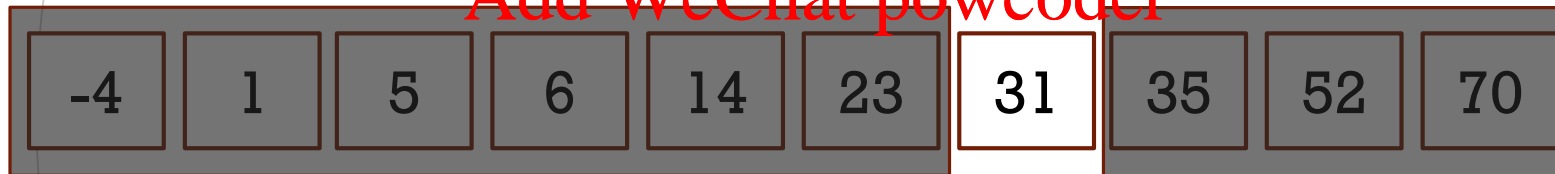
EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



EXAMPLE

- Search for 25
- Look at the middle element and compare

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

-4	1	5	6	14	23	31	35	52	70
----	---	---	---	----	----	----	----	----	----

EXAMPLE

- Search for 25
- There are no more elements in the list → the element is not there! Return -1.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

-4	1	5	6	14	23	31	35	52	70
----	---	---	---	----	----	----	----	----	----

IMPLEMENT BINARY SEARCH

- Idea: keep track of the left and right indices denoting the section of the list that needs to be searched.

<https://powcoder.com>

- What is the index of the element that we compare to the key as a function of the left and right indices?

BACK TO EXAMPLE

- Search for 25 (initialize left and right)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

-4	1	5	6	14	23	31	35	52	70
----	---	---	---	----	----	----	----	----	----

left = 0

right = 9

right = size - 1

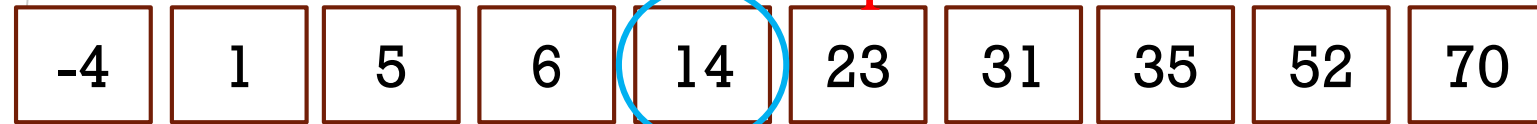
BACK TO EXAMPLE

- Search for 25
- Look at the middle element and compare (compute mid)

Assignment Project Exam Help

$\text{mid} = (\text{left} + \text{right}) / 2$
 $\text{mid} = 4$
<https://powcoder.com>

Add WeChat powcoder



left = 0

right = 9

BACK TO EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half (update left)

Assignment Project Exam Help

<https://powcoder.com>

mid = 4

Add WeChat powcoder



left = 5

right = 9

left = mid + 1

EXAMPLE

- Search for 25
- Look at the middle element and compare (compute mid)

Assignment Project Exam Help

<https://powcoder.com>

$$\text{mid} = (\text{left} + \text{right}) / 2 \\ = 7$$

Add WeChat powcoder



left = 5

right = 9

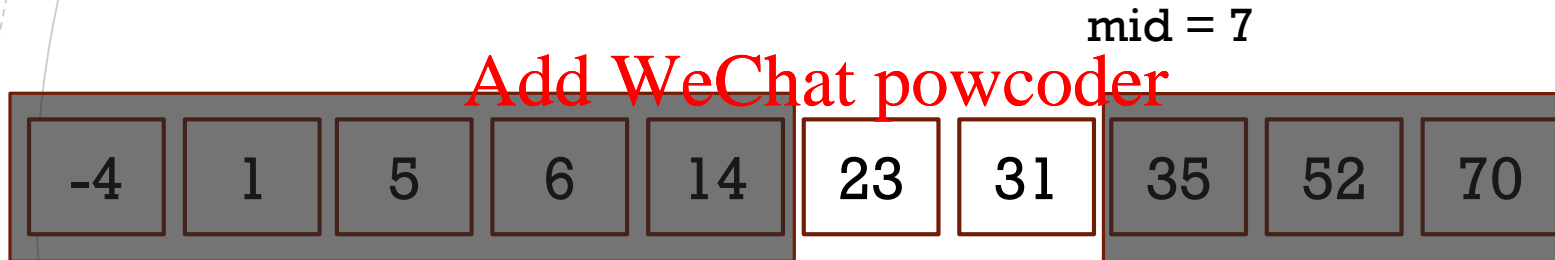
EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half (update right)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



left = 5 right = 6
 = mid - 1

EXAMPLE

- Search for 25
- Look at the middle element and compare (compute mid)

Assignment Project Exam Help

<https://powcoder.com>

mid = 5

Add WeChat powcoder



left = 5 right = 6

EXAMPLE

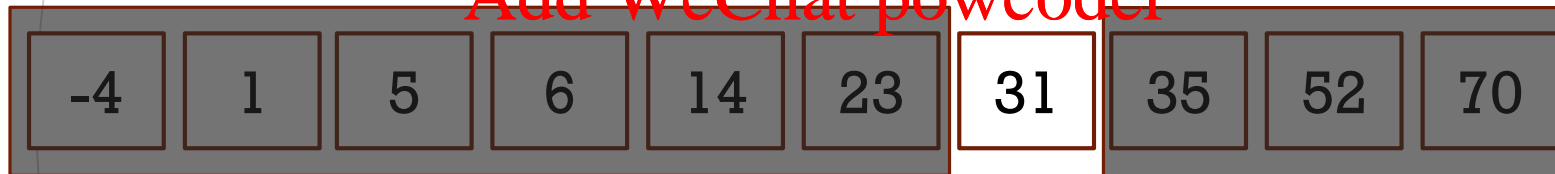
- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half (update left)

Assignment Project Exam Help

<https://powcoder.com>

mid = 5

Add WeChat powcoder



left = 6

right = 6

EXAMPLE

- Search for 25
- Look at the middle element and compare (compute mid)

Assignment Project Exam Help

<https://powcoder.com>

mid = 6

Add WeChat powcoder



left = 6
right = 6

EXAMPLE

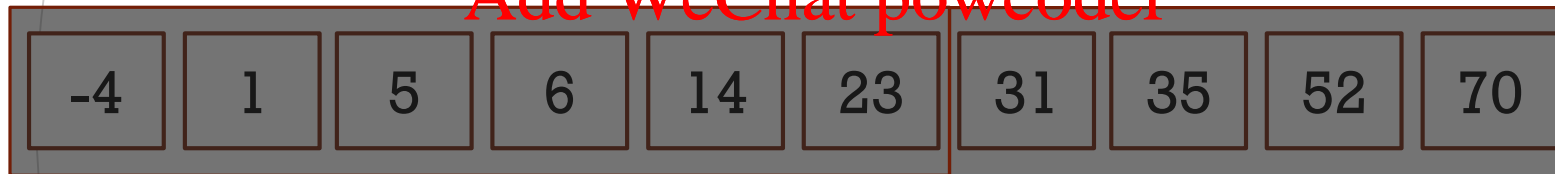
- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the list and keep searching on the other half (update right)

Assignment Project Exam Help

<https://powcoder.com>

mid = 6

Add WeChat powcoder



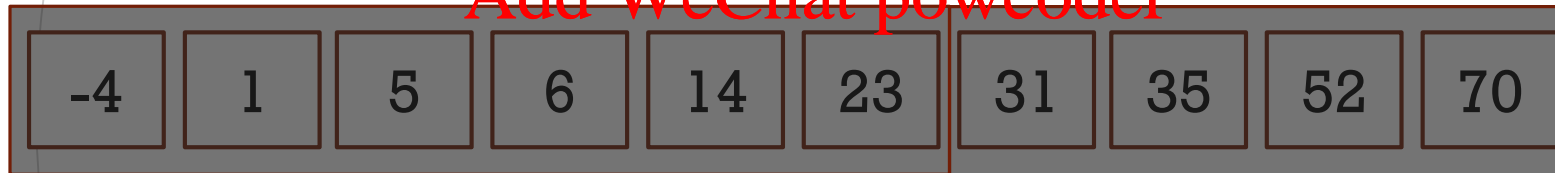
right= 5 left = 6

EXAMPLE

- Search for 25
- There are no more elements in the list ($\text{right} < \text{left}$)
→ the element is not there! Return -1.

<https://powcoder.com>

Add WeChat powcoder



right= 5 left = 6

BINARY SEARCH (ITERATIVE)

```
binarySearch(list, key){  
    left = 0  
    right = list.size() - 1  
    while(left <= right){  
        // initialize left and right  
        // until there are elements to search  
  
        ...  
  
        // key not in list  
    }  
    return -1  
}
```

Assignment Project Exam Help

<https://powcoder.com>

...

Add WeChat powcoder

BINARY SEARCH (ITERATIVE)

```
binarySearch(list, key){
    left = 0
    right = list.size() - 1
    while(left <= right){
        mid = (left + right)/2
        if(list[mid]==key)
            return mid
        else {
            // update either left or right
        }
    }
    return -1
}
```

initialize left and right

until there are elements to search

compute mid

compare element with key

key not in list

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

BINARY SEARCH (ITERATIVE)

```
binarySearch(list, key){
    left = 0
    right = list.size() - 1
    while(left <= right){
        mid = (left + right)/2
        if(list[mid]==key)
            return mid
        else {
            if(key<list[mid])
                right = mid -1
            else
                left = mid + 1
        }
    }
    return -1
}
```

initialize left and right

until there are elements to search

compute mid

compare element with key

update right

update left

key not in list

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

BINARY SEARCH (RECURSIVE)

```
binarySearch(list, key){  
    left = 0  
    right = list.size() - 1  
    while(left <= right){  
        mid = (left + right)/2  
        if(list[mid]==key)  
            return mid  
        else {  
            if(key<list[mid])  
                right = mid -1  
            else  
                left = mid + 1  
        }  
    }  
    return -1  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

What should change?

BINARY SEARCH (RECURSIVE)

```
binarySearch(list, key, left, right) {  
    while(left <= right) {  
        mid = (left + right) / 2  
        if(list[mid] == key)  
            return mid  
        else {  
            if(key < list[mid])  
                right = mid - 1  
            else  
                left = mid + 1  
        }  
    }  
    return -1  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Pass left and right as
parameters to the method

BINARY SEARCH (RECURSIVE)

```
binarySearch(list, key, left, right) {  
    if(left <= right) {  
        mid = (left + right) / 2  
        if(list[mid] == key)  
            return mid  
        else {  
            if(key < list[mid])  
                right = mid - 1  
            else  
                left = mid + 1  
        }  
    }  
    return -1  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Replace the while with an if
Add WeChat powcoder

BINARY SEARCH (RECURSIVE)

```
binarySearch(list, key, left, right) {  
    if(left <= right) {  
        mid = (left + right)/2  
        if(list[mid]==key)  
            return mid  
        else {  
            if(key<list[mid])  
                return binarySearch(list, key, left, mid-1)  
            else  
                return binarySearch(list, key, mid+1, right)  
        }  
    }  
    return -1  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add recursive calls

Add WeChat powcoder

OBSERVATIONS

Q: How many times through the while loop ? (iterative)
How many recursive calls? (recursive)

Add WeChat powcoder

A:

OBSERVATIONS

Q: How many times through the while loop ? (iterative)
How many recursive calls? (recursive)

Add WeChat powcoder

A: Worst case: the element cannot be found. Then, worst case is $O(\log n)$ where n is size of the list. Why? Because each time we are approximately halving the size of the list.



Coming Soon

Assignment Project Exam Help

In the next videos:

<https://powcoder.com>

- Quick sort

- Merge sort

- Recurrences

Add WeChat powcoder