

COMP 250

INTRODUCTORY SCIENCE

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Week 9-8 : Recursion (ch)

Add WeChat edu_assist_pro

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- More recursive Algorithms

Assignment Project Exam Help

- Decimal to Binar

<https://eduassistpro.github.io/>

- Power function

- Binary Search

Add WeChat edu_assist_pro

RECALL: DECIMAL TO BINARY (ITERATIVE)

ALGORITHM

Constructing Base 2 Expansions

Assignment Project Exam Help

pro n)

$k :=$ <https://eduassistpro.github.io/>

Whi $n > 0$

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

$n := n / 2$

$k := k + 1$

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

```
pro https://eduassistpro.github.io/  
If  $n$  Add WeChat edu\_assist\_pro  
    BinaryExpansion  
    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

- Inductive definition: <https://eduassistpro.github.io/>

- Base clause:

Add WeChat edu_assist_pro

$$x^0 = 1$$

- 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://eduassistpro.github.io/>
Add WeChat edu_assist_pro

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

POWER (x^n) – RECURSIVE

Assignment Project Exam Help

```
power(x, n) {
```

```
    if(n
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
    r
```

```
    } else {
```

```
        return x*powe
```

```
    }
```

```
}
```

POWER() – CAN WE DO BETTER?

More interesting approach using recursion:

Assignment Project Exam Help

x^1

<https://eduassistpro.github.io/>

x^9

Add WeChat edu_assist_pro

x^4

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

$$x^2 = x * x$$

POWER (x^n) – RECURSIVE 2

```
power( x, n) {  
    if (n == 0)  
        return 1;  
    else if (n ==  
        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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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:

$$x^n = x^{a_{k-1}2^{k-1} + \dots + a_12^1 + a_02^0}$$

<https://eduassistpro.github.io/>

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 [edu_assist_pro](https://eduassistpro.github.io/)

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; // 1 multiplication  
        n = n/2;  
    }  
    return result;  
}
```

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro
Let $n = (a_{k-1}, \dots, a_1, a_0)_2$
 $n) - 1$ iterations
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; // 1 multiplication  
        n = n/2;  
    }  
    return result;  
}
```

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro
Let $n = (a_k, \dots, a_1, a_0)_2$
 $n) + 1$ iterations
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; // 1 multiplication  
        n = n/2;  
    }  
    return result;  
}
```

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

n iterations
1 multiplication

EXAMPLE: x^{243}

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

Assignment Project Exam Help

Q: How many m

<https://eduassistpro.github.io/>
d?

Add WeChat edu_assist_pro

EXAMPLE: x^{243}

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

Assignment Project Exam Help

Q: How many m <https://eduassistpro.github.io/> d?

Add WeChat edu_assist_pro

A: Recursive method: $5*2 + 2*$

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

Q: How many m <https://eduassistpro.github.io/>
d?

Add WeChat edu_assist_pro

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 c <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

A: No. Why not ?

OBSERVATIONS

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

- x^2 has about ? digits
- x^3 has about ? digits
- :
- x^n has about ? digits.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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.
- <https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

We cannot assume that multiplication takes 'constant' time.

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

Assignment Project Exam Help

B <https://eduassistpro.github.io/> H

Add WeChat edu_assist_pro

SEARCHING A LIST

- Goal: find a given element in a list.

Assignment Project Exam Help

- Solution: go through <https://eduassistpro.github.io/> list and check whether the element is there (*linear search*)

Add WeChat edu_assist_pro

- 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.)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

BINARY SEARCH

- Inputs:
 - A sorted list.
 - The element we are looking for (the *key*)
- IDEA: First compare the middle of the list
 - If the key is less than the middle element, only need to search the first half of the list, so we continue search 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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

EXAMPLE

- Search for 25

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

-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://eduassistpro.github.io/>

Add WeChat edu_assist_pro



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half hing on the other half

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of the array, continuing on the other half

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



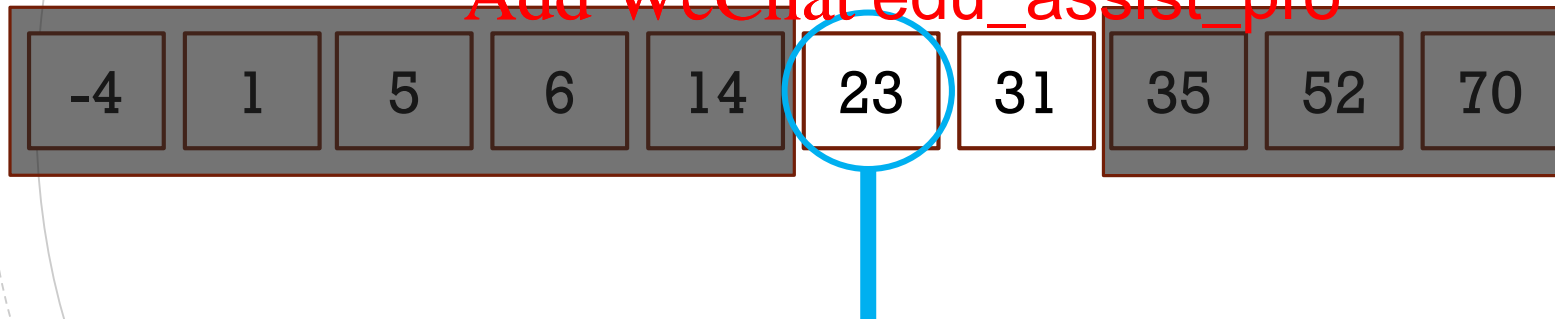
EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half hing on the other half

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



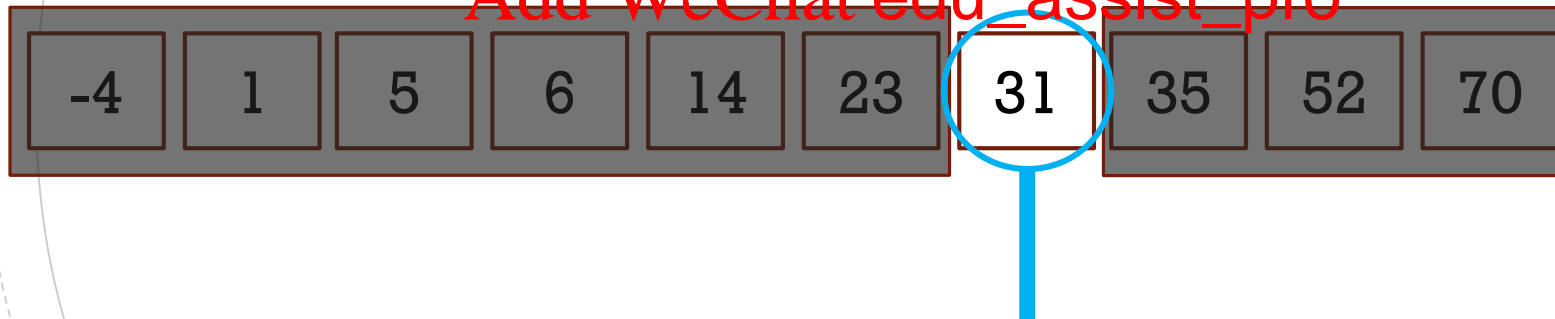
EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half hing on the other half

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

-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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

-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://eduassistpro.github.io/>

- What is the index of the element that is less than or equal 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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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

left = 0

right = 9

right = size - 1

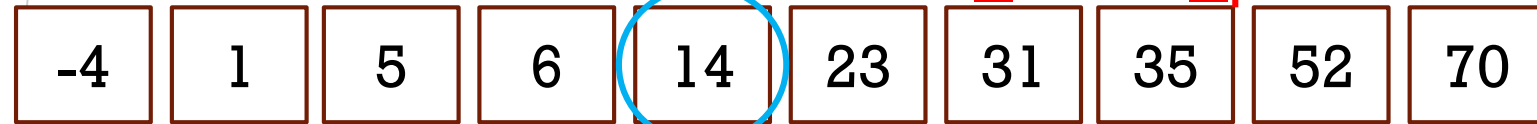
BACK TO EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



left = 0

right = 9

BACK TO EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



left = 5

right = 9

left = mid + 1

EXAMPLE

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

Assignment Project Exam Help

$\frac{\text{left} + \text{right}}{2}$
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



left = 5

right = 9

EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of t

Assignment Project Exam Help

n the other half (update right)

<https://eduassistpro.github.io/>

= 7

Add WeChat edu_assist_pro



left = 5 right = 6
= mid - 1

EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

mid

Add WeChat edu_assist_pro



left = 5 right = 6

EXAMPLE

- Search for 25
- Look at the middle element and compare
- If not equal: discard half of t

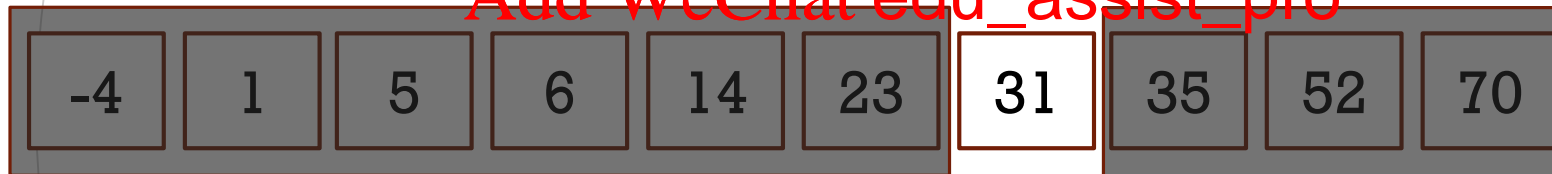
Assignment Project Exam Help

on the other half (update left)

<https://eduassistpro.github.io/>

mid

Add WeChat edu_assist_pro



left = 6

right = 6

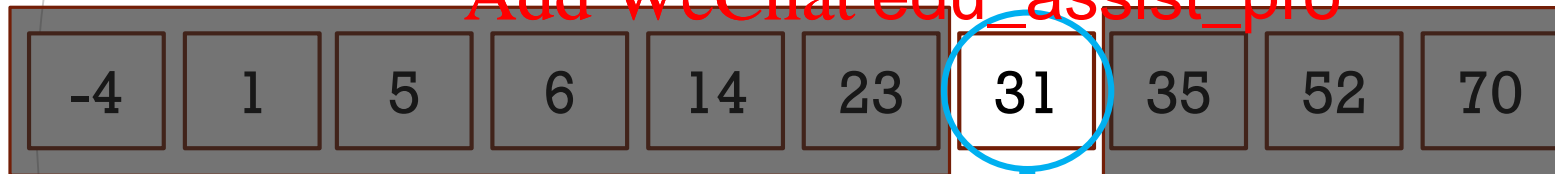
EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



left = 6
right = 6

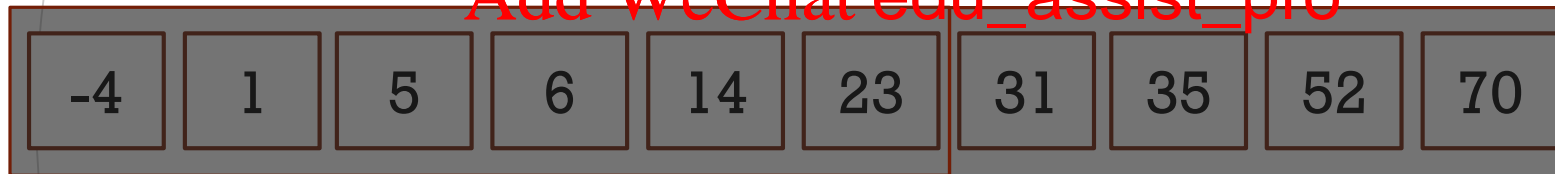
EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



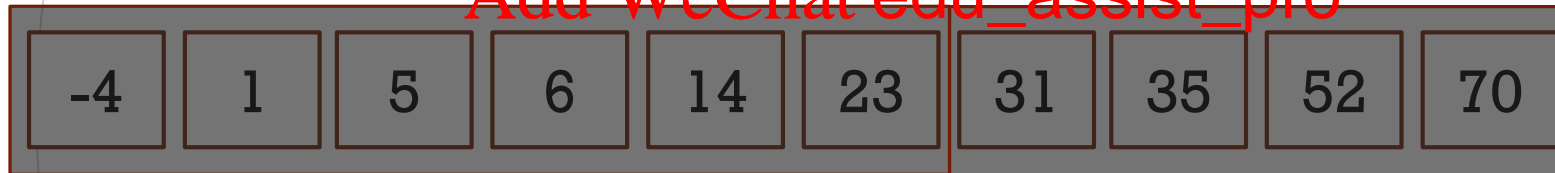
right= 5 left = 6

EXAMPLE

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

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



right= 5 left = 6

BINARY SEARCH (ITERATIVE)

```
binarySearch(list, key){  
    left = 0  
    right = list.size() - 1  
    while(left <= right){  
        ...  
    }  
    return -1  
}
```

initialize left and right
until there are elements to search
// key not in list

Assignment Project Exam Help

<https://eduassistpro.github.io/>

...

Add WeChat edu_assist_pro

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

element with key

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

key not in list

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

an element with key

// update right

// update left

// key not in list

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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://eduassistpro.github.io/>

Add WeChat edu_assist_pro 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://eduassistpro.github.io/>

Add WeChat edu_assist_pro
ft and right as
eters 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://eduassistpro.github.io/>

Add WeChat edu_assist_pro e the while with an if

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(left, mid-1)  
            else  
                return binarySearch(list, key, mid+1, right)  
        }  
    }  
    return -1  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

cursive calls

OBSERVATIONS

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

How many <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

A:

OBSERVATIONS

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

How many <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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

- Quick sort <https://eduassistpro.github.io/>
- Merge sort Add WeChat edu_assist_pro
- Recurrences