

check is Array is Sorted or not

Lecture 33: Recursion and Binary Search | Day-3 | 10 Days Recursion Challenge

2	4	6	9	11	13
---	---	---	---	----	----

function → isSorted
 ↘ Loops
 Recursion

for

```
bool isSorted (int arr[], int size)
```

```
{
```

```
    // base case
```

```
    if (size == 0 || size == 1)
        return true;
```

```
    if (arr[0] > arr[1]) // sorted nhi h
        return false;
```

```
    else
```

```
        bool ans = isSorted(arr+1, size-1);
```

ek bad
gaya then
size-1
ho jayega



```
#include<iostream>
using namespace std;

bool isSorted(int *arr, int size) {

    //base case
    if(size == 0 || size == 1 ){
        return true;
    }

    if(arr[0] > arr[1])
        return false;
    else {
        bool remainingPart = isSorted(arr + 1, size - 1 );
        return remainingPart;
    }
}

int main() {

    int arr[5] = {2,4,6,8,9};
    int size = 5;

    bool ans = isSorted(arr, size);

    if(ans){
```

lovebabbar
bar/Docume
Day3/" 66
babbar/Doc
ion Day3/"
Lovebabbar

arr+1

4.0 getSum :-

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

```
3
4 int getSum(int *arr, int size) {
5     //base case
6     if(size == 0) {
7         return 0;
8     }
9     if(size == 1)
10    {
11        return arr[0];
12    }
13
14    int remainingPart = getSum(arr+1, size-1);
15    int sum = arr[0] + remainingPart;
16
17
18
19
20 int main() {
21
22     int arr[5] = {2,4,9,9,9};
23     int size = 5;
24
25     int sum = getSum(arr, size);
26
27     cout << "Sum is " << sum << endl;
28
29     return 0;
```

lovebabbar/Doc
ion Day3/" 66
babbar/Doc
ion Day3/"
Lovebabbar

que :-

linear search :-

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

array :- key element = 6

0	1	2	3	4
3	5	1	2	6

found / not found

key = 2

return

arr

int arr[5] = [3 | 5 | 1 | 2 | 6]

main
int size = 5;
key = 2;

bool ans = linearsearch(arr, size, key)

Base case Agar array khali ho gya
and element nahi mila hai
if (size == 0)
return false;

if (arr[0] == key)
return true;

else.

{ bool-remaining part = linearsearch
return remaining (arr+1, size-1, key)
part }

[3 | 5 | 1 | 2 | 6 |]

key = 6

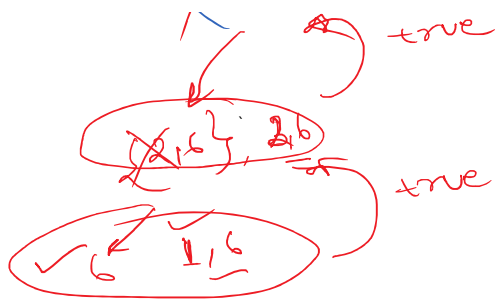
size = 5

{ ~~3~~ 5 1 2 6 }

{ ~~3~~ 1 2 6 }, 4, 6 → size.

{ ~~3~~ 1 2 6 }, 3, 6 → true.

{ ~~3~~ 1 2 6 }, 3, 6 → true.



if 6 ke jagah eight hota to.

Binary Search

Array sorted hone chahiye

0	1	2	3	4	5
2	7	6	10	14	16

$$\text{mid} = \frac{0+5}{2} = 2 \rightarrow 6$$

key 14

$14 > \text{mid}$

$$\text{mid} = \frac{3+5}{2} = 4 \rightarrow 14$$

$14 = 14 \rightarrow \text{return mid}$

#

0	1	2	3	4	5
2	7	6	10	14	18

key = 18

$$\frac{0+5}{2} = 2 \quad \text{mid} = 2$$

$\text{arr}[2] = 6$

if ($\text{arr}[\text{mid}] < \text{key}$)

Search right wale half me.

else

search karo left wale half me

Search in right half

$$s = \text{mid} + 1$$

search in left half

$$e = \text{mid} - 1$$

binary search (arr, start, end, key)

, Right

binary (arr, mid + 1, end, key)

→

1

left

binary (arr, start, end-1, key)

→ element not found.

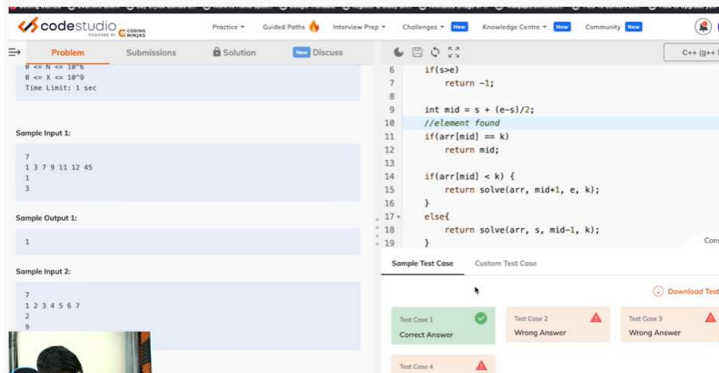
base case ① if (s > e) → return -1

② // element found.

if (arr[mid] == k)

return true;

②



#DSABusted #LoveBabbar
Lecture 34: Recursion with Strings | Day-4 | 10 Day Recursion Challenge

ques:-

i/p:- "a b c d e"
o/p:- "e d c b a"

② i/p:- "babbar"
o/p:- "rabbab"

~~e d c b a~~ , (n-1)

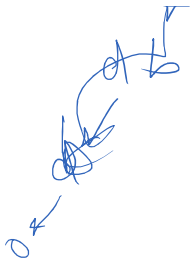
~~d c d b~~ {

if (str.length == 0)
return

c d b

reverse(i, j, str).

base case

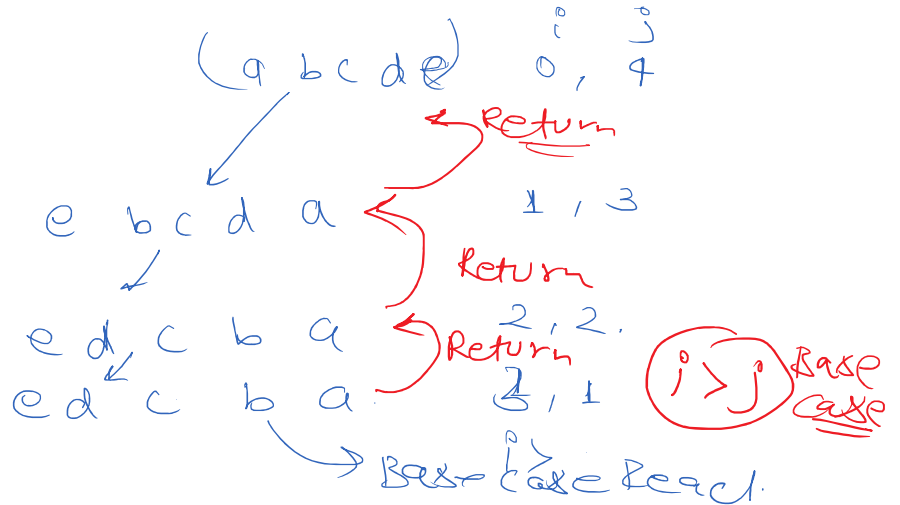


base case
 if ($i > j$)
 return;
 swap ($s[i], s[j]$)
 $i++$
 $j--$

reverse (i, j, str)

```
void reverse(string str, int i, int j) {
    //base case
    if (i > j)
        return;
    swap(str[i], str[j]);
    i++;
    j--;
    void reverse(BidirectionalIterator
        _BidirectionalIterator __first,
        _BidirectionalIterator __last)
    //Recursion
    reverse();
}

int main() {
    string name = "babbar";
}
```



que 2 :- check palindrome :-

string: abba

reverse string = a b b a

string == reverse string

bool checkPalindrom (string str, int i, int j)

```
{
    // Base case.
    if (i > j)
        return true;
    if (str[i] != str[j])
        return false;
    // Recursive call-
    else return checkPalindrom (str, i+1, j-1);
}
```

```
1 #include <iostream>
2 using namespace std;
3 bool checkPalindrom (string str, int i, int j) {
4     // Base case.
5     if (i > j)
6         return true;
7     if (str[i] != str[j])
8         return false;
9     // Recursive call-
10    else return checkPalindrom (str, i+1, j-1);
11 }
```

```

1 #include <iostream>
2 using namespace std;
3
4 bool checkPalindrome(string str, int i, int j) {
5
6     //base case
7     if(i >= j)
8         return true;
9
10    if(str[i] != str[j])
11        return false;
12    else {
13        //Recursive call
14        return checkPalindrome(str, i+1, j-1);
15    }
16 }
17
18 int main() {
19
20    string name = "BookkooB";
21    cout << endl;
22
23    bool isPalindrome = checkPalindrome(name, 0, name.length()-1);
24
25    if(isPalindrome)
26        cout << "It's a Palindrome";
27    else
28        cout << "It's not a Palindrome";
29
30    return 0;
31 }

```

que 2 i/p a=3

$a^b \rightarrow$ b is even $\rightarrow a (a^{\frac{b}{2}} \times a^{\frac{b}{2}})^2$

$a^b \rightarrow$ b is odd $\rightarrow 2^9 = 2 \times 2^4 \times 2^4$

$2^9 \Rightarrow 2 \times 2^4 \times 2^4 \Rightarrow 2 \times (2^4)^2$

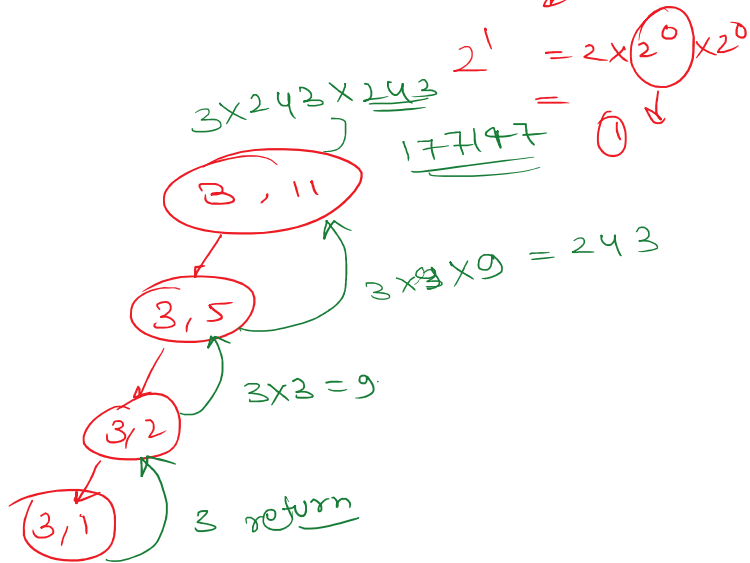
$2^4 \Rightarrow 2^2 \times 2^2 \Rightarrow (2^2)^2$

$2^2 = 2 \times 2 \Rightarrow (2)^2$

```

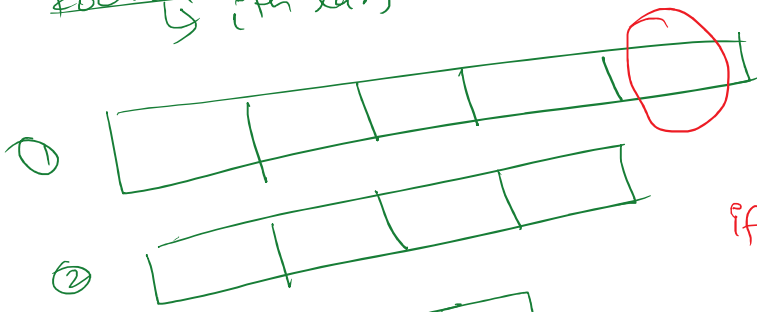
1 //power.cpp
2
3 int power(int a, int b) {
4     //base case
5     if(b == 0)
6         return 1;
7
8     if(b == 1)
9         return a;
10
11    //RECURSIVE CALL
12    int ans = power(a, b/2);
13
14    //if b is even
15    if(b%2 == 0) {
16        return ans * ans;
17    }
18    else {
19        //if b is odd
20        return a * ans * ans;
21    }
22 }
23
24 int main() {
25
26    int a, b;
27    cin >> a >> b;
28    cout << endl;
29
30    cout << power(a, b);
31 }

```



Bubble Sort

Bound i^{th} $\rightarrow i^{th}$ largest (Right place)

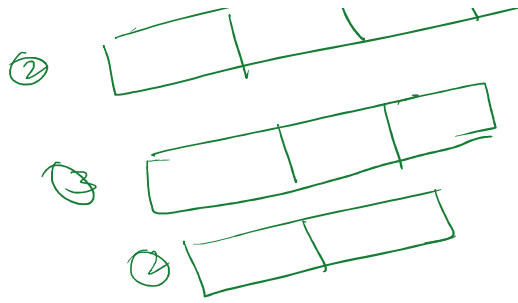


Base case

if size == 0 || size == 1

sorted

return;



if size - 1 == 0
sorted
return;

for(int i=0; i<n; i++)
if (arr[i] > arr[i+1])
swap(arr[i], arr[i+1])

}
sort array (arr, n-1)

```

1 #include<iostream>
2 using namespace std;
3
4 void sortArray(int arr, int n) {
5
6     //Base case - already sorted
7     if(n == 0 || n == 1) {
8         return ;
9     }
10
11     //1 case solve karlia - largest element ko end me rakh dega
12     for(int i=0; i<n-1; i++){
13         if(arr[i] > arr[i+1]){
14             swap(arr[i], arr[i+1]);
15         }
16     }
17
18     sortArray(arr, n-1);
19 }
20
21 int main() {
22
23     int arr[5] = {2,5,1,6,9};
24     sortArray(arr,5);
25
26     for(int i=0; i<5; i++){

```

HW:- ~~Bubble sort~~
insertion & selection
sort using

Recursion