

CS5343.001: Homework #1

Due on September 3, 2016 at 11:59pm

Professor Greg Ozbirn

Lizhong Zhang

Problem 1

Suppose your calculator only did base 10 logarithms. Write an expression to compute log base 2 of 2048 using only log base 10.

Solution

$$\log_2 2048 = \frac{\log_{10} 2048}{\log_{10} 2} = 11$$

Problem 2

Express the following summation in closed form (an expression that can be directly computed from k). (Refer to slide 11)

$$3 + 5 + 7 + 9 + \dots + 2k + 1$$

Solution

$$\sum_{i=1}^k (2i + 1) = \frac{(3 + 2k + 1)k}{2} = k(k + 2)$$

Problem 3

Proof by counterexample.

Prove that the following statement is false: $n^3 > 2^n$ for any $n \geq 1$.

Justification

Suppose $n = 1$, $n^3 = 1$ and $2^n = 2$. Apparently, $n^3 < 2^n$.

Thus, the statement is false.

Problem 4

Proof by contradiction.

Prove that the following statement is true: the square of an even number is also even.

Justification

Assume the statement is false: give an even number $x = 2k$, then $x^2 = (2k)^2 = 4k^2$ is an odd number. This indicate that an even number equals to an odd number, which is impossible.

Thus, the statement is true. The square of an even number is also even.

Problem 5

Induction proofs.

Part One

Prove by induction:

$$\sum_{i=1}^n i^3 = \frac{(n^2)(n+1)^2}{4}$$

Justification

Base case: $n = 1$.

$$\sum_{i=1}^1 i^3 = \frac{(1^2)(1+1)^2}{4} = 1$$

So, it is true for $n = 1$.

Induction step: Assume it is true for k , that $\sum_{i=1}^k i^3 = \frac{(k^2)(k+1)^2}{4}$.

Show true for $k + 1$;

$$\begin{aligned} \sum_{i=1}^{k+1} i^3 &= \sum_{i=1}^k i^3 + (k+1)^3 \\ &= \frac{k^2(k+1)^2}{4} + (k+1)^3 \\ &= \frac{k^2(k+1)^2 + 4(k+1)^3}{4} \\ &= \frac{k^2(k+1)^2 + 4(k+1)^2(k+1)}{4} \\ &= \frac{(k+1)^2(k^2 + 4k + 4)}{4} \\ &= \frac{(k+1)^2(k+2)^2}{4} \end{aligned}$$

Conclusion: by induction, the statement holds true for all n .

Part Two

Prove by induction:

$$n^2 - n \text{ is even for any } n \geq 1.$$

Justification

Base case: $n = 1$, then, $n^2 - n = 1^2 - 1 = 0$ is even. So, it is true for $n = 1$.

Induction step: Assume it is true for k , that $k^2 - k$ is even, so $k^2 - k = 2a$.

Show true for $k + 1$;

$$\begin{aligned} (k+1)^2 - (k+1) &= (k+1)((k+1) - 1) \\ &= (k+1)k \\ &= k^2 + k \\ &= k^2 - k + 2k \\ &= 2a + 2k \end{aligned}$$

We know $2a$ is even and $2k$ is even for all $n \geq 1$.

So, $2a + 2k$ is even.

Conclusion: by induction, the statement holds true for all $n \geq 1$.

Problem 6

Recursion

Note: You can use Java or pseudocode for these. If pseudocode then the logic must be complete and easy to understand.

Part One

Write a recursive function that when passed a value n displays.

$n (n - 1) (n - 2) (n - 3) \dots 0 \dots (n - 3) (n - 2) (n - 1) n$

for example, if passed 5 displays:

5 4 3 2 1 0 1 2 3 4 5

Solution

The Java code is as follow:

```
1 import java.util.Scanner;
2
3 public class Recursion1
4 {
5     public static void print(int n)
6     {
7         System.out.print(n + " ");
8         if (n > 0){
9             print(n - 1);
10            System.out.print(n + " ");
11            return;
12        }
13        else if (n == 0)
14            return;
15    }
16
17    public static void main(String args[])
18    {
19        Scanner num = new Scanner(System.in);
20        System.out.println("Enter a number: ");
21        int n = num.nextInt();
22        print(n);
23    }
24 }
```

Part Two

Write a recursive function that receives an array of integers and a position as parameters and returns the count of odd numbers in the array. Let each recursive call consider the next integer in the array.

Solution

The Java code is as follow:

```
1 import java.util.Scanner;
2
3 public class Recursion2
4 {
5     public static void main(String args[])
```

```
6      {
7          int arr[] = {0,1,2,3,4,5,6,7,8,9};
8          for (int b = 0; b < arr.length; b++){
9              System.out.print(arr[b]);
10         }
11         Scanner num = new Scanner(System.in);
12         System.out.print("\nEnter a number: ");
13         int n = num.nextInt();
14         int i = countOdd(arr, n);
15         System.out.println("The count of odd numbers in the array is " + i);
16     }
17
18     public static int countOdd(int []array, int n)
19     {
20         if (n < array.length){
21             if (array[n] % 2 != 0)
22                 return 1 + countOdd(array, n+1);
23             else
24                 return countOdd(array, n+1);
25         }
26         else
27             return 0;
28     }
29 }
```

Problem 7

Suppose there exists a generic Java class named Pair with type parameter T that stores two objects with get and set methods for each. Write the statements necessary to create an object of type Pair with String as its type parameter, and use the set methods to set the two strings, then the get methods to retrieve them for printing. Note that you do not need to write the Pair class itself.

Solution

The Java code is as follow:

```
1 public static void main(String args[])
2 {
3     Pair<String> cr = new Pair<>();
4     cr.set1("This is the first sentence!");
5     cr.set2("This is the second sentence!");
6     String myMethod1 = cr1.get1();
7     System.out.println(myMethod1);
8     String myMethod2 = cr2.get2();
9     System.out.println(myMethod2);
10 }
```