**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Midterm Exam**

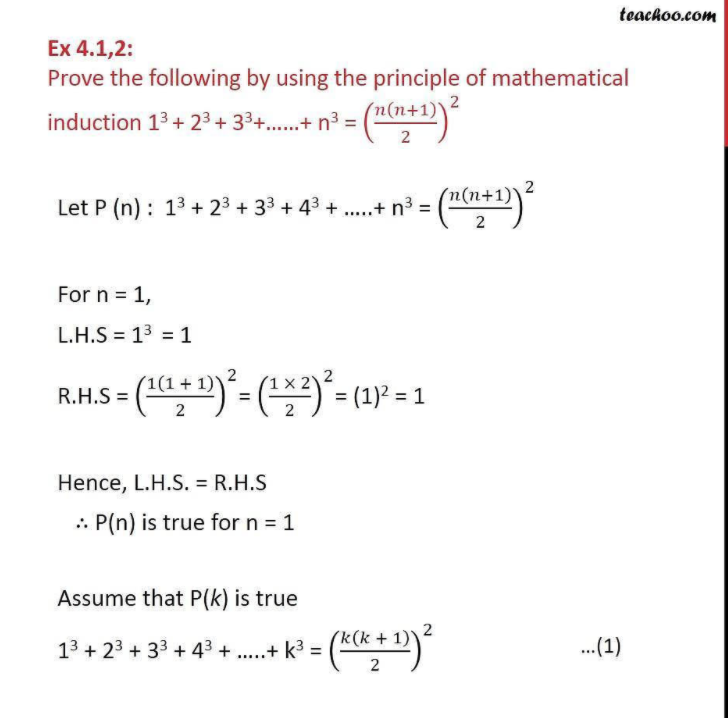
**There are 8 questions and 45 points + 5 bonus points. Time Limit : 2 hours**

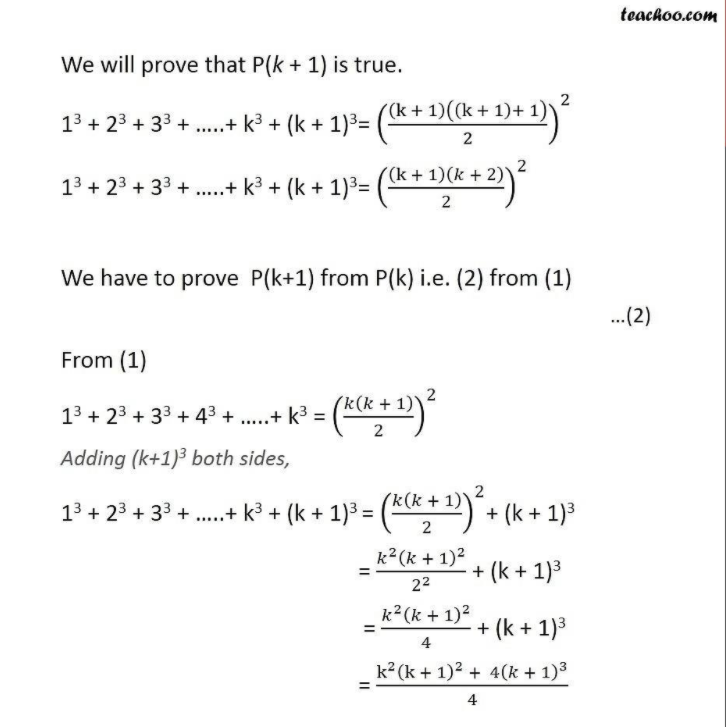
**Question 1 [6 Points]**

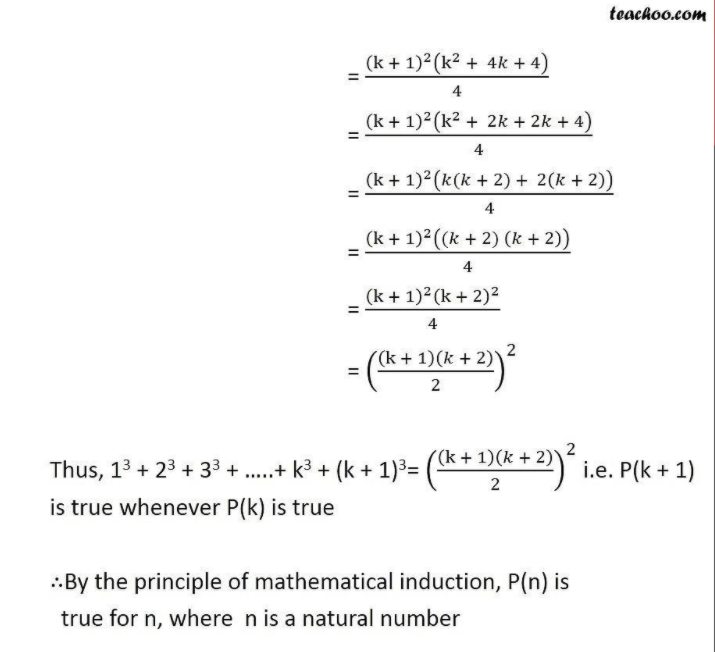
*(This could be the easiest or the most difficult problem. So, if you are not sure, try at the end)*

1. What is 13 + 23 + 33 + …+ n3?
2. Prove your “Guess” using mathematical induction.

**Ans**







**Question 2 [6 Points] Fill in the blank and prove any TWO of the following:**

1. a + (a + d) + (a + 2d) + … + (a + (n-1)d) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

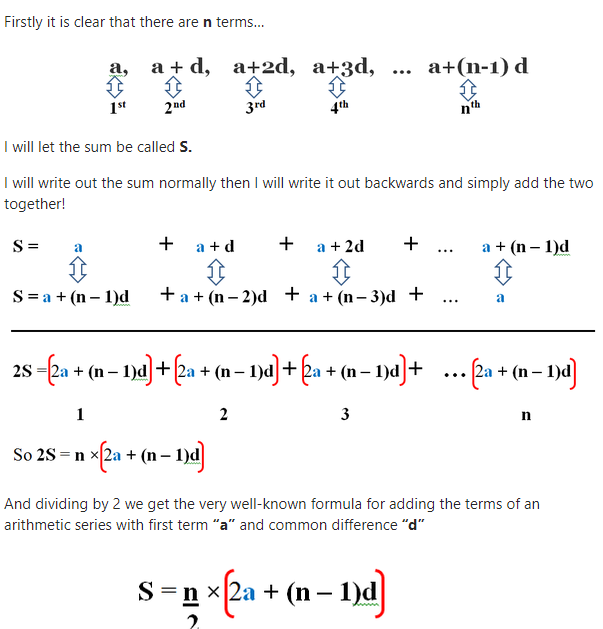
**ANS**

**P(n)= a + (n-1)d**

Arithmetic series formula

S(n)= [n\*(2\*first +(n-1)\*difference]/2

= [n(2a + (n-1)d)]/2



1. 1/2 + 1/4 + 1/8 + … = \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ANS**

**Geometric series**

**P(n)=a(1-rn-1)**

**-> S(n) = [a(1-rn)]/(1-r) where r= next/previous terms = [¼]/[1/2]**

**S =** 1/2 + 1/4 + 1/8 + … r = ½

**1/2S =** 1/4 + 1/8 +….

**S-1/2S = ½**

**S=1**

The [proof](https://everything2.com/title/proof) that the [formula](https://everything2.com/title/formula) **S**n**=(a(1-r**n**))/(1-r)** sums *n* terms of a geometric series is as follows, where *r* is the ratio of the series (which is between -1 and 1) and *a* is the [first term](https://everything2.com/title/first+term) in the series.

Sn=ar0+ar1+ar2+...+arn-2+arn-1  
(r)Sn=r(ar0+ar1+ar2+...+arn-1)  
r\*Sn=ar1+ar2+ar3+...+arn-1+arn  
a+r\*Sn=a+ar1+ar2+ar3+...+arn-1+arn  
a+r\*Sn=Sn+arn  
Sn-r\*Sn=a-arn  
Sn(1-r)=a(1-rn  
Sn=(a(1-rn))/(1-r)

1. a + ax + ax2 + … +ax(n -1) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

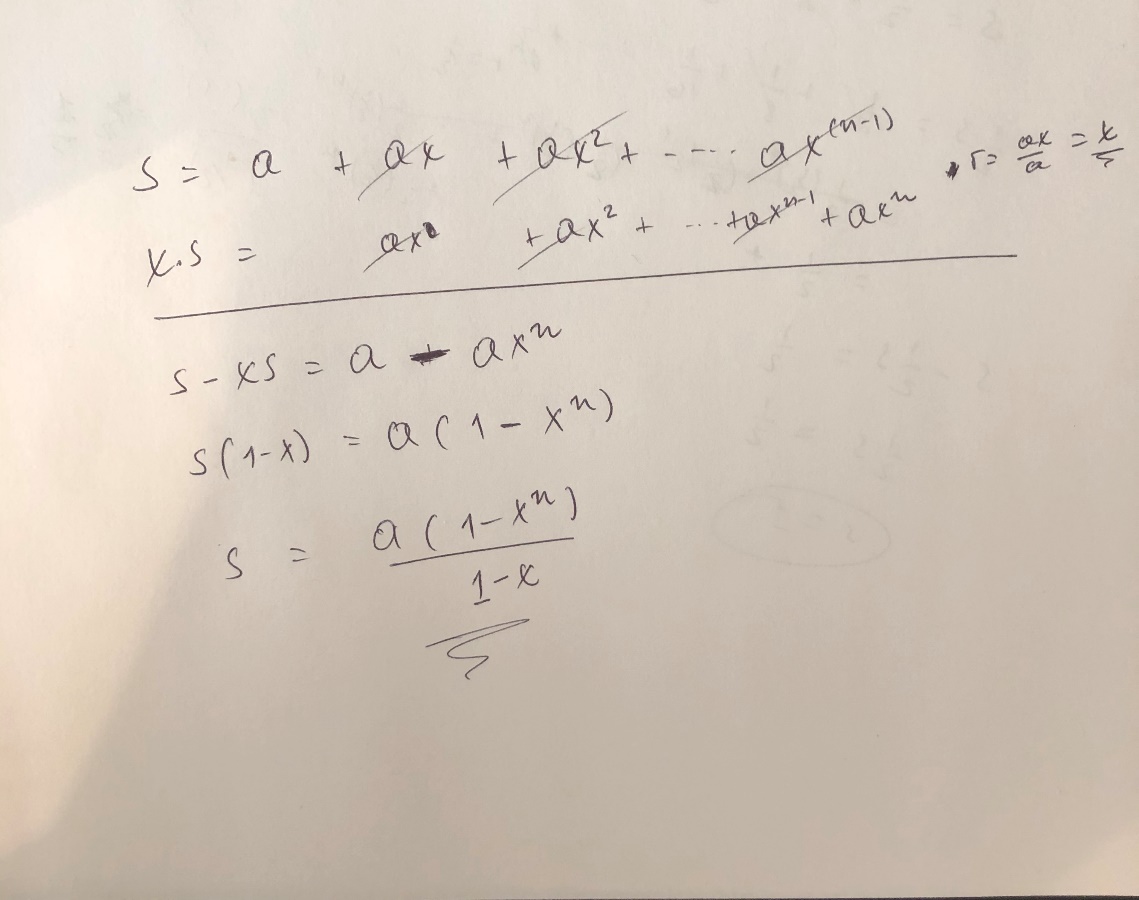
**ANS**

**r=x**

**S = [a(1-rn)]/[1-r]**

**S= [a(1-xn)]/[1-xn]**

**Proof:**



**Question 3 [6 Points] Order these functions in the increasing order of complexity.**

n1/3 log n, n1/4 log n, n1/3, log3 n, log5 n, log n

**ANS**

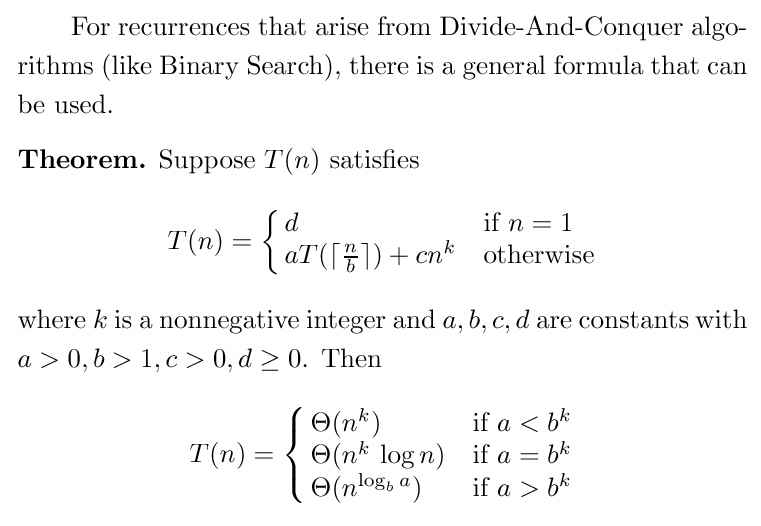
log n, log3 n, log5 n, n1/3,n1/4 log n, n1/3 log n

**Question 4 [6 Points]**

1. What is Euclid’s algorithm to compute GCD?
2. Prove it will always produce the correct answer.
3. What is the “Best case?”

**Question 5 [6 Points]**

1. If an array has 100 integers, what is the expected number of inversions?
2. Give three sorting algorithms you know that are inversion bound. What are their average time complexities?
3. Give two sorting algorithms you know that are NOT inversion bound. What are their worst time complexities?

****

**The Master Theorem**

**Question 6 [6 Points]**

Dr. MT was designing an algorithm. He found that he can solve the problem by DAC (Divide And Conquer). He has the following two options:

Algorithm A: Divide the problem into 7 sub problems each of size n/3. Further there is only an O(1) cost in combining the solutions of those 7 sub-problems to obtain the solution of the problem.

Algorithm B: Divide the problem into 4 sub problems each of size n/4. Further, there is an O(n) cost in combining the solutions of those 4 sub-problems to obtain the solution of the problem.

1. What is the time complexity of the Algorithm A.
2. What is the time complexity of the Algorithm B.

**Question 7 [6 Points]**

A bucket has 5 Red, 3 White and 2 Blue balls. Miss Jane Austin was picking a ball at random. After observing the color, she puts back in the bucket. Answer next three questions based on this fact.

1. On an average, how many times Miss Jane has to pick the ball to get a Blue ball?
2. On an average, how many times Miss Jane has to pick the ball to get a 10 Blue balls?
3. On an average, how many Red balls Miss Jane has to pick before she can get 10 Blue balls?

**Question 8[1 + 2 + 5 Points]**

**Remember the following which you have worked on in the Lab W1D5?**

**c** function: 1 to add.

3k to resize (if k >0. Note: k is the size of the “completely filled array”)

**ĉ** function: 7 to add. (customer is willing to pay for add)

0 to resize (customer do not want to pay for resizing. It is not his/her concern)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item #** | **Operation** | **Cost for us** | **Customer paid** | **Profit** | **Balance** |
| 1 | Add | We assume we start with 1 slot. We add 1 item at the cost of 1. | 7 | 6 | 6 |
| 2 | Add | 3 to resize  (We have two slots)  1 to add | 7 | 6 | 3  9 |
| 3 | Add | 6 to resize  (We have 4 slots)  1 to add | 7 | 6 | 3  9 |
| 4 | Add | 1 to add | 7 | 6 | 15 |
| 5 | Add | 12 to resize  (We have 8 slots)  1 to add | 7 | 6 | 3  9 |
| 6 | Add | 1 to add | 7 | 6 | 15 |
|  |  |  |  |  |  |

**Let us make a small change. Instead of “doubling” the size during the resize operation, you are “tripling” the size during the resize operation.**

Please show me the changes required. In particular,

1. What is c(resize) ? (Previously it was 3k)
2. What is the **minimum** value possible for ĉ(add)? (Previously 7)

Keep c(add) as 1 and ĉ(resize) as 0. (Previous values. No change)

1. Create a table similar to the one shown with “new values”. Show 10 rows.

**Have a nice weekend!**