1. Prove the following sum is $\Theta(n^3)$. You are not allowed to use a formula for the sum value. Rather, you must use simple manipulations of the terms to obtain appropriate upper and lower bounds.

$$\sum_{i=1}^{n} (i^2)$$

(a) Prove $O(n^3)$

(b) Prove $\Omega(n^3)$
2. (a) Consider the following recurrence for Mergesort algorithm for the case when $n$ is a power of 2. Prove by induction that the solution is $f(n) \leq An \log n$ and find the constant $A$.

$$f(n) = \begin{cases} 
2f(n/2) + n - 1, & n \geq 2 \\
0, & n = 1.
\end{cases}$$

(b) Use master-theorem to find the solution form of the following recurrence in terms of constants $A$ and $B$, and determine the constants. Assume $n$ is a power of 2.

$$T(n) = \begin{cases} 
4T(n/2) + 2n, & n \geq 2 \\
0, & n = 1.
\end{cases}$$
3. Given a sorted array of $n$ elements, $A[0..n-1]$, and a constant $C$. We want to determine if there is a pair of elements $A[i]$ and $A[j]$, $i \neq j$, such that $A[i] + A[j] = C$. (We want a Boolean function which returns a TRUE/FALSE answer.)

(a) Outline how this problem may be solved by using the binary-search algorithm, $BS(A, left, right, key)$. (Do not give the code for BS. It is a given function, which you call.) Analyze the time complexity of this approach.

(b) Describe a more efficient $O(n)$ algorithm to solve this problem. Give the pseudo-code. Explain how the algorithm works, and provide a numerical illustration.
4. This problem deals with computing $x^n$, where $x$ is real and $n$ is integer, using an efficient number of multiplications. More exactly, we want an efficient algorithm that uses $O(\log n)$ operations of only the following: real multiplications; integer add/subtract; integer multiply/divide by 2; integer mod 2; and no other arithmetic operations.

(a) For the special case when $n$ is a power of 2 ($n = 2^k$, $k \geq 1$), show how to compute $x^n$. What is the number of real multiplications?

(b) Now consider the general case when $n$ is any integer $\geq 2$, Write a recursive algorithm (pseudocode), POWER(x,n), to compute $x^n$.

(c) Let $f(n)$ be the worst-case number of real multiplications used by the above recursive algorithm. Write a recurrence for $f(n)$ and find the solution.
5. Recall the heap algorithm BUILD-HEAP starts with an array $A[1..n]$ of $n$ random elements and establishes a valid min-heap of $n$ elements.

(a) What is the time complexity of the algorithm BUILD-HEAP?

(b) Write the pseudocode for a recursive version of BUILD-HEAP. You can make calls to PUSH-DOWN as a given function. (Don’t give the code for pushdown.)

(c) Illustrate the working of BUILD-HEAP for the following array. Show the initial tree, and the result after each PUSHDOWN is completed.

$$A[1..13] = (13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1).$$