Department of Mathematics
Indian Institute of Technology Madras
Course Number : MA5910: Data Structures and Algorithms
Home Assignment Number : 1

---

1. Write an algorithm for generating all the permutation of $(1, 2, 3, \ldots, n)$ exactly once.
2. Let $P(x) = \sum_{k=0}^{n} a_k x^k$ be a polynomial of degree $n \geq 0$, where $a_0, a_1, \ldots, a_n$ are given constants. Write an algorithm that evaluates $P(x)$, for a given value x, that performs at most $n+1$ multiplications.
3. Let $A[1 \ldots n]$ be an array of n distinct integers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A. What is the relationship between the running time of insertion sort and the number of inversions in the input array? Justify your answer. Also, give an algorithm that determines the number of inversions in any permutation on n elements in $\Theta(n \log n)$ worst-case time.
4. Write C-programs for sorting n-given integers by using Bubble-Sort, Merge-Sort, Quick-Sort and Randomized Quick-Sort methods and compare their running time over a large set of inputs.
5. Prove the following statements:
   i. For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.
   ii. $max\{f(n), g(n)\} = \Theta(f(n)+g(n))$, where $f(n)$ and $g(n)$ are asymptotically nonnegative functions.
6. $o(g(n)) = \{f(n) \mid$ for any constant $c > 0, \exists$ a constant $n_0 > 0,$ such that $0 \leq f(n) < g(n), \forall n \geq n_o\}$.
   $\omega(g(n)) = \{f(n) \mid$ for any constant $c > 0, \exists$ a constant $n_0 > 0,$ such that $0 \leq cg(n) < f(n), \forall n \geq n_o\}$.

   By using the above definitions prove that $o(g(n)) \cap \omega(g(n)) = \phi$.
7. Find the solution of the following recurrence relations:
   i. $T(n) = T(\lceil \frac{n}{2} \rceil) + 1$
   ii. $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + n$
   iii. $T(n) = 2T(\lfloor \frac{n}{2} \rfloor + 37) + n$
   iv. $T(n) = 2T(\sqrt{n}) + 1$
   v. $T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + n$
   vi. $T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + cn$, where c is a constant.
   vii. $T(n) = T(n-a) + T(a) + cn$, where $a \geq 1$ and $c > 0$.
   viii. $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.
8. Use master theorem to find the exact solution of the following recurrence relations:
   i. $T(n) = 9T(\frac{n}{3}) + n$.
   ii. $T(n) = 4T(\frac{n}{2}) + n^3$.
   iii. $T(n) = T(\frac{n}{2}) + \Theta(1)$.
   iv. $T(n) = 4T(\frac{n}{2}) + n^2 \log n$.