

	Worst	Best	Average
1. OPERATION	Sort Array (unsorted)	Sort Array (sorted)	Sort Array (best case)
Add	$4 = O(1)$	$T(n) = 8n + 8 = O(n)$	$O(n)$
find Min	$5n + 4 = O(n)$	$T(n) = O(1)$	$O(n)$
find max/median	$6n + 4 = O(n)$	$T(n) = O(n)$	$O(n)$
mean	$4n + 4 = O(n)$	$O(n) T(n) = O(n)$	$O(n)$
contains	$4n + 2 = O(n)$	$T(n) = 4n + 2 = O(n)$	$O(n)$

2. void symDifference (bitset result, Bitset A, Bitset B) {

 // Bitset result = new Bitset();

 for (int i = 0; i < MAX; i++) {

 if (A[i] == 1 && B[i] == 0) result[i] = 1;

 if (B[i] == 1 && A[i] == 0) result[i] = 1;

 }

 return result;

3. Polynomial multiply (Polynomial p1, Polynomial p2)

Polynomial result = new Poly (result → size);

node ptr pA = p1 → next;

node ptr pB = p2 → next;

while (pA != NULL) {

 pA = pA → next;

 while (pB != NULL) {

 result = pA → coef * pB → coef;

 result = pA → max degree + pB → max degree;

 pB = pB → next;

 pA = pA → next;

return result;

$$T(n) = 13n + 4$$

$$= O(n)$$

4. int dummy(int i) {

 int sum, i, j, k;

 sum = 0;

 for (i = 1; i <= n; i++) {

 for (j = 1; j <= i; j++) {

 for (k = 1; k <= j; k++) {

 sum = sum + k;

 } return sum;

$$\sum_{j=1}^i [5j^2 + 3] + 5i + 2$$

$$= \frac{5i^3}{3} + 3i + 2i + 2$$

$$= \frac{5i^3}{3} + 5i + 2$$

$$\sum_{i=1}^n \left[\frac{5i^3}{3} + 5i + 2 \right] + 2n + 3$$

$$\frac{5n^4}{12} + \frac{5n^2}{2} + 4n + 3$$

$$T(n) = \frac{15n^4}{12} + \frac{5n^2}{2} + 4n + 3$$

$$T(n) = \frac{15n^4}{12} + \frac{5n^2}{2} + 4n + 3$$

$$= O(n)$$

$$O: \frac{15n^4}{12} + \frac{5n^2}{2} + 4n + 3 < O(n) \quad n \rightarrow \infty \quad \left(\frac{15n^4}{12} + \frac{5n^2}{2} + 4n + 3 \right) > 2n \quad \text{C.O.D.} \quad \text{Hence } O(n)$$

C.O.D. is not possible because of the above concept C.O.D.

$$\frac{15n^4}{12} + \frac{5n^2}{2} + 4n + 3 < O$$

$$\therefore T(n) \neq O(n)$$

$$T(n)$$

IMPOSSIBLE $\therefore O(n)$ is not $O(n)$

$$\therefore T(n) \text{ is not } O(n)$$

```

5. int level(treeptr t) {
    if (t == NULL) {

```

```

    5. int level(treeptr t) {
        int level, dlevel;
        if (t != NULL) {
            if (t->left == t->parent) {
                return level;
            }
            dlevel = level(t->right) + level++ ;
            if (dlevel != 0) {
                return dlevel;
            }
            dlevel = level(t->left) + level++ ;
            return dlevel;
        }
        return 0;
    }

```

BEST CASE : IF NO ROOT, RUNNING TIME WILL BE CONSTANT

AVERAGE CASE : IF MOST ITEMS ARE IN ONE TREE OR LEFT SIDE

Worst case : IF AVERAGE NO. OF ITEMS

Worst case : THE BIGGER THE NUMBER OF ELEMENTS, THE WORSE IT IS

$$T(n) = O(n^2)$$