Provide big-O notations:

1. $n^2 + 3n - 7$ → $O(n^2)$ → $O(n^2)$
2. $5n^3 - 3n^2 + 8n - 2$ → $O(n^3)$ → $O(n^3)$
3. $\log(n) - 2 + 3n$ → $O(\log n)$ → $O(\lg n)$
4. $n\log(n)$ → $O(n \log n)$ → $O(\#n \lg n)$

screw the notation i wrote in pen (fixed)

```
1. for (int i = 1; i <= n; i++) {          cost: c1    time: n+1
     for (int j = 1; j <= n; j++) {        c2          n*(n+1)
       x = 2 * y;   *header runs extra     c3          n^2
       z = x--;                            c4          n^2
     }
   }
```

$T(n) = (n+1)*c_1 + (n^2+n)*c_2 + n^2*c_3 + n^2*c_4$

Notation: $O(n^2)$

```
2. for (int i = 1; i <= n; i++) {          cost: c1    time: (n+1)
     j = n;                                c2          #n
     while (j >= 1) {                      c3          n*(n+1)
       x++;                                c4          n^2
       j = j / 2;                          c5          n^2
     }
   }
```

$T(n) = (n+1)*c_1 + n*c_2 + (n^2+n)*c_3 + n^2*c_4 + n^2*c_5$

Notation: $O(n^2)$

```
3. i = 1;                  cost: c1    time: 1
   while (i != n)          c2          (n+1)
     i++;                  c3          n
```

$T(n) = c_3*n + c_2*(n+1) + c_1$

Notation: $O(n)$

```
4. for (int i = 1; i <= n; i++) {          cost: c1    time: n+1
     if (i == 1 || i == n || i == 3) {     c2          n
       for (int j = 1; j <= n; j++) {      c3          3*n+1
         x++;                              c4          3*n
       }
     }
   }
```

$T(n) = (n+1)*c_1 + n*c_2 + (3n+3)*c_3 + 3n*c_4$

Notation: $O(n)$