

Task Description:

Task-1: Count the number of primitive operations executed below and determine the best & the worst cases: (1 points)

Algorithm: arrayMin(A, n)
 $currentMin \leftarrow A[0]$
 $i \leftarrow 1$
while $i \leq n - 1$ **do**
 if $currentMin \geq A[i]$ **then**
 $currentMin \leftarrow A[i]$
 $i \leftarrow i + 1$
return $currentMin$

Best<
 $3 \rightarrow \text{ewttgpvO kp} \rightarrow \text{'C']2_}$
 $3 \rightarrow \text{k? '3}$
 $n \rightarrow \text{k'>? 'p/' '3}$
 $n - 1 \rightarrow \text{ewttgpvO kp} < \text{'C]k_}$
 $n - 1 \rightarrow i = i + 1$
 $3 \rightarrow \text{tgwtp'ewttgpvO kp}$
Vqvcn3n + 1

Y qtuw<
 $3 \rightarrow \text{ewttgpvO kp} \rightarrow \text{'C']2_}$
 $3 \rightarrow \text{k? '3}$
 $p \rightarrow \text{k'>? 'p/' '3}$
 $p - 1 \rightarrow \text{ewttgpvO kp} > \text{'C]k_}$
 $p \rightarrow \text{'3} \rightarrow \text{ewttgpvO kp} \rightarrow \text{'C]k_}$
 $p \rightarrow \text{'3} \rightarrow \text{k? 'k- '3}$
 $3 \rightarrow \text{tgwtp'ewttgpvO kp}$
Vqvcn6p

Task-2: Determine the Big-O notation for: (3 points)

a) $2 + n(2 + 3n)$ $O(n^2)$

b) $n + 2(n + 3n)n + \frac{n}{2}$ $O(n^2)$

c) $n^3 \log n + 2n + 1 + 3n^2 + n(\log n)^2$ $O(n^3 \log n)$

Task-3: Determine the Complexity Of The Following Small Functions: (6 points)

a) **for** ($i = \text{sum} = 0; i < n; i++$) $O(n)$
 $\text{sum} += a[i];$

b) **for** ($i = 0; i < n; i++$) $O(n^2)$
 for ($j = 0; j < n; j++$)
 $a[i][j] = i*j;$

c) **for** ($i = n; i \geq 1; i--$) $O(n^2)$
 for ($j = i; j \leq n; j++$) /* Note that the value of the inner loop variable (j) */
 ... /* depends on the value of the outer loop variable (i) */

d) `for (i = 1; i <= n; i++)` $O(n)$
`for (j = i; j <= i; j++)` `/* Note that the value of the inner loop variable (j) */`
`...` `/* depends on the value of the outer loop variable (i) */`

e) `for (i = 0; i < n; i++)` $O(n \log n)$
`for (j = n; j > 1; j/=2)`
`...`

f) `int factorial (int n)` $O(n)$
`{`
`if (n <= 1)`
`return 1;`
`else`
`return n * factorial(n-1);`
`}`