**Algorithm-1**

|  |  |  |
| --- | --- | --- |
| Step | Cost of each execution | Total # of times executed |
| 1 | 1 | 1 |
| 2 | 1 | n+1 |
| 3 | 1 | Σi=1 to n(i+1) |
| 4 | 1 | Σi=1 to n(i) |
| 5 | 1 | Σi=1 to n Σj=1 to i(j+1) |
| 6 | 6 | Σi=1 to n Σj=1 to i(j) |
| 7 | 7 | Σi=1 to n(i) |
| 8 | 2 | 1 |

Multiply col.1 with col.2, add across rows and simplify

T1(n) = 1 + n + 1 + Σi=1 to n(i+1) + Σi=1 to n (i)+ Σj=1 to i Σi=1 to n (j+1) + 6(Σj=1 to i Σi=1 to n (j)) + 7(Σi=1 to n (i)) + 2

= 4 + n + n(n+1)/2 + n + n(n+1)/2 + (n(n+1)/2)\*n + (n(n+1)/2)\*2 + 6(n(n+1)/2)\*n + 7(n(n+1)/2

= 4 +2n + (n^2 + n)/2 + (n^2 +n)/2 + ....

T1(n) = O(n^3)

**Algorithm-2**

|  |  |  |
| --- | --- | --- |
| Step | Cost of each execution | Total # of times executed |
| 1 | 1 | 1 |
| 2 | 1 | n+1 |
| 3 | 1 | n |
| 4 | 1 | Σi=1 to n (i+1) |
| 5 | 6 | Σi=1 to n (i) |
| 6 | 7 | Σi=1 to n (i) |
| 7 | 2 | 1 |

Multiply col.1 with col.2, add across rows and simplify

T2(n) = 1 + n + 1 + n + Σi=1 to n (i+1) + 6(Σi=1 to n (i)) + 7(Σi=1 to n (i)) + 2

= 3 + 3n + 7(n(n+1))

= 2 +10n + 7n^2

= O(n^2)

**Algorithm-3**

|  |  |  |
| --- | --- | --- |
| Step | Cost of each execution | Total # of times executed in any single recursive call |
| 1 | 4 | 1 |
| 2 | 11 | 1 |
| Steps executed when the input is a base case:1 or 2 | | |
| First recurrence relation: T(n=1 or n=0) = 4 for n = 0 and 11 for n = 1 | | |
| 3 | 5 | 1 |
| 4 | 2 | 1 |
| 5 | 1 | (n/2) + 1 |
| 6 | 6 | n/2 |
| 7 | 7 | n/2 |
| 8 | 2 | 1 |
| 9 | 1 | (n/2) + 1 |
| 10 | 6 | n/2 |
| 11 | 7 | n/2 |
| 12 | 4 | 1 |
| 13 | 4 | (cost excluding the recursive call) 1 |
| 14 | 5 | (cost excluding the recursive call) 1 |
| 15 | 17 | 1 |
| Steps executed when input is NOT a base case: steps 1 to 15 | | |
| Second recurrence relation: T(n>1) = 56 + 14n | | |
| Simplified second recurrence relation (ignore the constant term): T(n>1) =14n | | |

Solve the two recurrence relations using any method (recommended method is the Recursion Tree). Show your work below:



T3(n) = 14nlogn+14n

T3(n) = O(nlogn)

**Algorithm-4**

|  |  |  |
| --- | --- | --- |
| Step | Cost of each execution | Total # of times executed |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | n+1 |
| 4 | 10 | n |
| 5 | 7 | n |
| 6 | 2 | 1 |

Multiply col.1 with col.2, add across rows and simplify

T4(n) =1 + 1 + n + 1 + 10n + 7n + 2

=18n+5

T4(n)= O(n)



Based on the data provided by the graph, all the curves seem to follow the theoretical time needed to run each of the algorithms.