**Algorithm-1**

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

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

T1(n) = θ(n3)

**Algorithm-2**

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

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

T2(n) = θ(n2)

**Algorithm-3**

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

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

T3(n) = O(nlogn)

**Algorithm-4**

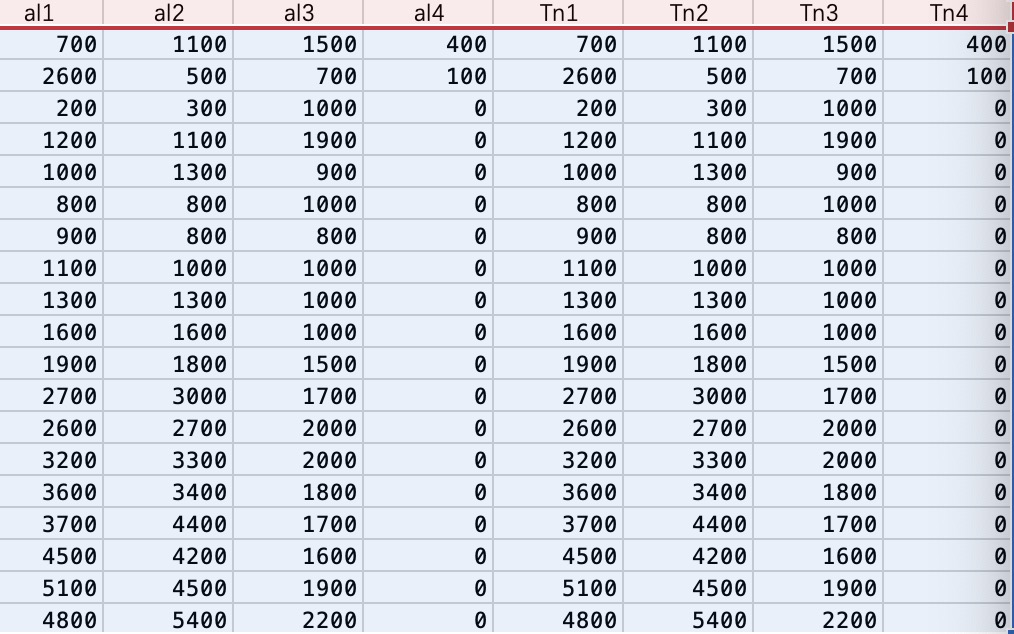
|  |  |  |
| --- | --- | --- |
| Step | Cost of each execution | Total # of times executed |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | n+1 |
| 4 | 8 | n |
| 5 | 5 | n |
| 6 | 2 | 1 |

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

T4(n) = 1+1+n+1+8n+5n+2

=14n+5

=θ(n)



图片包含 文字

描述已自动生成