Time complexities | Ethan Wilkes & Cole Spencer

Algorithm 1

|  |  |  |
| --- | --- | --- |
| Step | Cost of each Execution | Number of times executed |
| 1 | 1 | 1 |
| 2 | (Q - L + 1) \* (1 + (U-L+1) + 1) | Q-P+1 |
| 3 | (1 + (U-L+1) + 1) | Q-L+1 |
| 4 | 1 | Q-L+1 |
| 5 | 1 | U-L+1 |
| 6 | 1 | (Q-P+1) \* (Q-L+1)(U-L+1) |
| 7 | 1 | (Q-P+1) \* (Q-L+1) |
| 8 | 1 | 1 |

T(n)= O((Q-P)(Q-L)(U-L))

Complexity = n^3

Algorithm 2

|  |  |  |
| --- | --- | --- |
| Step | Cost of each Execution | Number of times executed |
| 1 | 1 | 1 |
| 2 | 2 | Q-P+1 |
| 3 | (1 + (U-L+1) + 1) | Q-P+1 |
| 4 | 1 | Q-L+1 |
| 5 | 1 | (Q-P+1) \* (Q-L+1) |
| 6 | 1 | (Q-P+1) \* (Q-L+1) |
| 7 | 1 | 1 |

T(n)= O((Q-P)(Q-L))

Complexity = n^2

Algorithm 3

|  |  |  |
| --- | --- | --- |
| Step | Cost of each Execution | Number of times executed |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | 1 |
| 4 | 1 | 1 |
| 5 |  | n/2 + 1 (or M-L) |
| 6 | n (5 and 6) | n/2 inner loop |
| 7 | 1 | 1 |
| 8 | 1 | 1 |
| 9 |  | n/2 + 1 (or U-M) |
| 10 | N (9 and 10) | n/2 |
| 11 | 1 | 1 |
| 12 | 1 | 1 |
| 13 | T(n/2) | 1 |
| 14 | T(n/2) | 1 |
| 15 | 1 | 1 |

T(n)=2T(n/2) + 2n + 9

T(n)=2T(n/2)+n (ignoring constants)

Complexity = nLogn

Algorithm 4

|  |  |  |
| --- | --- | --- |
| Step | Cost of each Execution | Number of times executed |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | Q-P+1 |
| 4 | 2 | Q-P+1 |
| 5 | 1 | Q-P+1 |
| 6 | 1 | 1 |

T(n)=O(Q-P) Complexity = n

A screenshot of a computer

Description automatically generated with medium confidence

For algorithms one and two, we have several nested for loops. In algorithm one there are 3 nested for loops. This means the time complexity comes out to n3. Algorithm 2 has two nested for loops which makes the time complexity n2. The last two algorithms don’t have nested loops, but because of the if statements + for statement which takes n, the total complexity would be nlogn. The fourth and final algorithm consists only of a for loop, this gives the time complexity n. The actual time for each algorithm matched closely. Ignoring constants can make it a negligible amount off, but it generally matched with the time complexities that were calculated going line by line.