**CIS 360 Lab #2: Implementing and Analyze Algorithms**

**Name \_\_\_Tanmay Sonawane\_\_\_\_\_\_\_\_ ID \_\_\_\_01996456\_\_\_\_\_\_\_\_**

**Task A.** Implement the three algorithms in Section 1.4 in Zybooks:

* MaxsubSlow (Figure 1.4.2)
* MaxsubFaster (Figure 1.4.3)
* MaxsubFastest (Figure 1.4.4)

*Note that, for all three algorithms, A must contain both positive and negative numbers.*

Text, letter

Description automatically generated

Text, letter

Description automatically generated

*Note: S is a list of numbers, not a variable. Si denotes the i-th element, which will contain the prefix sum of A[i] (i.e. the sum of all elements in A up to A[i]). The first element, S0, is given a value of 0 because prefix sum of A[0] is 0, since A[0] does not exist.*

Text, letter

Description automatically generated

*Note: M is a list of numbers, not a variable. Mi denotes the i-th element, which is the prefix maximum for A[i] (i.e. the largest subarray of A that can be yielded using only elements up to A[i]). The first element, M0, is given a value of 0 because the prefix maximum of A[0] is 0, since A[0] does not exist.*

**Task B.** For each algorithm , perform experiments with 10 different values for n, and measure the running time (i.e. the number of primitive operations, **not** the system runtime). Select appropriate n values so the running time is measurable and the change in running time is meaningful. Once you have yielded the runtimes, use the tables below to determine which of the following functions best approximates the running time for each algorithm: n, n2, or n3.

*Hint: For each value of n, calculate the ratio between the algorithm’s running time and the function. As n grows infinitely large, the function whose ratio converges to a nonzero value closest to 1 will be the best approximation.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | T1(n)  MaxsubSlow | T1(n)/n | T1(n)/(n2) | T1(n)/(n3) |
| 100 | 171703 | 1717.03 | 17.1703 | 0.171703 |
| 200 | 1353403 | 6767.015 | 33.835 | 0.1691 |
| 300 | 4545103 | 15150.343 | 50.501 | 0.1683 |
| 400 | 10746803 | 26867.0 | 67.167 | 0.1679 |
| 500 | 20958503 | 41917.0 | 83.834 | 0.1676 |
| 600 | 36180203 | 60300.338 | 100.50 | 0.1675 |
| 700 | 52498351 | 74997.644 | 107.1395 | 0.16743 |
| 800 | 85653603 | 107067.00 | 133.833 | 0.1672 |
| 900 | 121905303 | 135450.336 | 150.50 | 0.16722 |
| 1000 | 167167003 | 167167.003 | 167.167 | 0.1671 |
|  |  |  |  |  |
| n🡪infinite | Predict this: | Infinity | Infinity | 0.16 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | T2(n)  MaxsubFaster | T2(n)/n | T2(n)/(n2) | T2(n)/(n3) |
| 100 | 5153 | 51.53 | 0.5153 | 0.005153 |
| 200 | 20303 | 101.51 | 0.507 | 0.002 |
| 300 | 45453 | 151.51 | 0.5050 | 0.0016 |
| 400 | 80603 | 201.5075 | 0.5037 | 0.0012 |
| 500 | 125753 | 251.506 | 0.5030 | 0.0010 |
| 600 | 180903 | 301.505 | 0.5025 | 0.0008 |
| 700 | 246053 | 351.504 | 0.5021 | 0.0007 |
| 800 | 321203 | 401.503 | 0.5018 | 0.0006 |
| 900 | 406353 | 451.503 | 0.5016 | 0.0005 |
| 1000 | 501503 | 501.50 | 0.501 | 0.0005 |
|  |  |  |  |  |
| n🡪infinite | Predict this: | infinity | 0.5 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | T3(n)  MaxsubFastest | T3(n)/n | T3(n)/(n2) | T3(n)/(n3) |
| 100 | 204 | 2.04 | 0.0204 | 0.000204 |
| 200 | 404 | 2.02 | 0.010 | 5.05e-05 |
| 300 | 604 | 2.013 | 0.0067 | 2.237e-05 |
| 400 | 804 | 2.01 | 0.0050 | 1.2562e-05 |
| 500 | 1004 | 2.008 | 0.0040 | 8.032e-06 |
| 600 | 1204 | 2.006 | 0.0033 | 5.574e-06 |
| 700 | 1404 | 2.005 | 0.0028 | 4.0932e-06 |
| 800 | 1604 | 2.005 | 0.0025 | 3.132e-06 |
| 900 | 1804 | 2.004 | 0.0022 | 2.474e-06 |
| 1000 | 2004 | 2.004 | 0.002004 | 2.004e-06 |
|  |  |  |  |  |
| n🡪infinite | Predict this: | 2 | 0 | 0 |

Based on the experimental data, which of the three functions (n, n2, or n3) is the best estimation of the time complexity T(n) of each algorithm?

Answer:

T1(n) = \_\_\_\_\_n^3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

T2(n) = \_\_\_\_\_n^2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

T3(n) = \_\_\_\_\_n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.