CPSC 1160 – Week 7 Homework

Student: Lana Cambeses

QUESTION 1: Count =, -, >=, --, += operators

(a) Operators count table or list

|  |  |  |
| --- | --- | --- |
| Operator | Count | Reason |
| = | 2 | int t = 0;  int i = n - 1 |
| - | 1 | int i = n - 1 |
| >= | n + 1 | i >= 0  (true n times and false once) |
| -- | n | Loop runs n times, and i-- is inside the loop |
| += | n | Loop runs n times, and += is inside the loop |

(b) Derivation of the total count function T(n)

T(n) = 2 + 1 + n + 1 + n + n

T(n) = 3n + 4

(c) Final Big O complexity and short reasoning

O(n) -> From the T(n), we drop non-dominant terms, as well as constant multiple of the dominant term.

QUESTION 2: Count >=, <, +=, --, ++ operators

(a) Operators count table or list

|  |  |  |
| --- | --- | --- |
| Operator | Count | Reason |
| >= | n + 1 | for (int i = n - 1; i >= 0; i--) (true n times and false once) |
| < | (n/2) + 1 | for (int j= 0; j < n / 2; j++)  (true n/2 times and false once) |
| += | n + (n/2) | It happens n times inside the first loop, and then it happens n/2 inside the second loop |
| -- | n | It happens n times inside the first loop |
| ++ | n/2 | It happens n/2 times inside the second loop |

(b) Derivation of the total count function T(n)

T(n) = n + 1 + (n/2) + 1 + n + (n/2) + n + (n/2)

T(n) = (3n/2) + 3n + 2

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

(c) Final Big O complexity and short reasoning

O(n) -> From the T(n), we drop non-dominant terms, as well as constant multiple of the dominant term (and constant divisor too).

QUESTION 3:

(a) Operators count table or list

|  |  |  |
| --- | --- | --- |
| Operator | Count | Reason |
| = | 1 + (n – 1) + (n – 1) + (n(n-1)/2) + (n – 1) | int i = 1  (It happens once at the start of the first loop)  int currentElement = list[i]; (This line happens n – 1 times because it’s how many times the I loop iterates)  k = i – 1  (This line happens n – 1 times because it’s how many times the I loop iterates)  list[k + 1] = list[k]; (Assuming the worst case and the conditions are always true, this happens I times. But as I keeps on changing at a constant rate of 1, we need to use an arithmetic series logic to get the total sum of the iterations. Since the loop goes from 0 to n – 1, the formula is applied as: the sum of each pair 1 + (n – 1) times the amount of pairs that is (n – 1)/2, which is (n(n-1)/2))  list[k + 1] = currentElement; (This happens n - 1 times) |
| < |  |  |
| [] |  |  |
| >= |  |  |
| > |  |  |

(b) Derivation of total count function T(n)

(c) Final Big O complexity and short reasoning

QUESTION 4: Count << operators

|  |  |  |
| --- | --- | --- |
| Operator | Count | Reason |
| << | 1st loop: outer loop is some log sort; inside loop is GS |  |

(b) Derivation of total count function T(n)

(c) Final Big O complexity and short reasoning

QUESTION 5: Count =, <=, [], > operators.

|  |  |  |
| --- | --- | --- |
| Operator | Count | Reason |
| = |  |  |
| <= |  |  |
| [] |  |  |
| > |  |  |

(b) Derivation of total count function T(n)

(c) Final Big O complexity and short reasoning