Name: Breona Pizzuta Date: 9/18/24

Pledge: “I pledge my honor that I have abided by the Stevens Honor System”

Give the asymptotic complexity of the following functions. Choose the most appropriate notation from among , , and . Give only a single answer for each question (giving more than one answer will result in a zero for that question).

1. **void** **function1**(**int** n) {

**for** (**int** i = 1; i <= n; i++) { //n times

**for** (**int** j = i; j <= n; j += 2) { // (n- i)/2 times

cout << "\*";

}

}

}

Answer:

1. **void** **function2**(**int** n) {

**int** count = 0;

**for** (**int** i = 1; i \* i \* i <= n; i++) { //cubic iterations n^1/3 times

count++;

}

cout << count;

}

Answer:

1. **void** **function3**(**int** n) {

**int** count = 0;

**for** (**int** i = 1; i \* i <= n; i++) { //n^1/2 times

**for** (**int** j = 1; j + n/2 <= n; j++) { // n/(n/2) times

**for** (**int** k = 1; k <= n; k \*= 2) { // log 2 n

count++;

}

}

}

cout << count;

}

Answer: wrong function should be 3/2 not 1/2

1. **void** **function4**(**int** n) {

**int** count = 0;

**for** (**int** i = n/2; i <= n; i++) { //n/2 times

**for** (**int** j = 1; j <= n; j \*= 2) { // log 2 n times

**for** (**int** k = 1; k <= n; k \*= 2) { // log 2 n times

count++;

}

}

}

cout << count;

}

Answer:

1. **void** **function5**(**int** n) {

**if** (n % 2 == 0) { //only if n is even it wont run. Otherwise n times

**return**;

}

**for** (**int** i = 1; i <= n; i++) { // n times

**for** (**int** j = 1; j <= n; j++) { // n times

cout << "\*";

**break**;

}

}

}

Answer:

1. **void** **function6**(**int** n) {

**int** count = 0;

**for** (**int** i = 1; i <= n/2; i++) { //n/2 times

**for** (**int** j = 1; j <= n/3; j++) { // n/3 times

**for** (**int** k = 1; k <= n/4; k++) { // n/4 times

count++;

}

}

}

cout << count;

}

Answer:

1. **void** **function7**(**int** n) {

**for** (**int** i = 1; i <= n; i++) { //n times

**for** (**int** j = 1; j <= n; j += i) { // sum of n/i times; sum is 1/I

**//**(harmonic series)

cout << "\*";

}

}

}

Answer: :

1. **void** **function8**(**int** n) {

**int** i = 1, s = 1;

**while** (s <= n) {

i++; // keeps adding until n>s. Series adding incr. by 1

s += i;

cout << "\*"; // i(i+1) / 2 = s stops when i^2 >n

}

}

Answer:

1. Processing Arrays
   1. Suppose you have an unsorted array of integers of length and want to sum all the elements inside it. What is the running time of your algorithm?
      1. because you need to count all n elements to the sum.
   2. Suppose you have an unsorted array of integers of length and want to determine if all the values inside are positive. What is the running time of your algorithm?
      1. because you will stop as soon as you hit a negative.
   3. Suppose you have a sorted array of integers of length and want to determine the median value. What is the running time of your algorithm?
      1. because you can directly access the element in the middle.
2. True or False: Answer: TRUE

Then, if true, prove it by giving integer values for the required constants , , and . Show your work in detail. Choose the tightest values possible for the and constants. If false, show a contradiction.

We need to show that

For the lower bound we can say = 5 because when

For the upper bound we can say = 6 because when

Therefore: ; ;