CS385Q: Algorithms
Fall 2022
Homework1

Asymptotic Notations

Due on: 09/27/2022 11:59 PM

1. (5 Points) Consider the following pseudo-code and analyze the best and the worst-case running time of this by providing the form of the function the running time takes in each of the cases (Analyze statement-by-statement and use the length of the array A as n):

```
Algorithm 1 AdjacentDuplicates(A)
  for $i = 0$ to $A.length - 1$ do
    if $A[i] == A[i+1]$ then
      return 1
    end if
  end for
  return 0
```

2. (10 Points) Consider the following pseudo-code and analyze the best and the worst-case running time of this by providing the form of the function the running time takes in each of the cases (Analyze statement-by-statement and use the length of the array A as n):

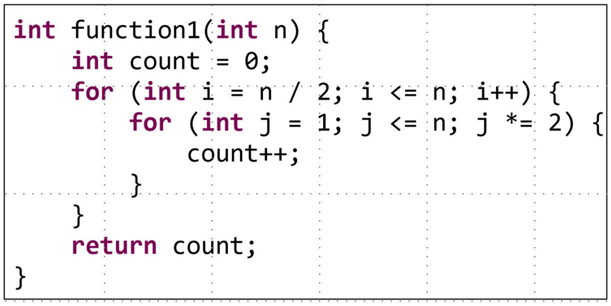
```
Algorithm 2 Linear-Search(A,v)
  $i = NIL$
  for $j = 1$ to $A.length$ do
    if $A[j] = v$ then
      $i = j$
      return $i$
    end if
  end for
  return $i$
```

3. (5 points) Find the upper bound for $f(n) = n^4 + 10n^2 + 5$. Prove your answer by giving values for the constants c and n_0. Choose the smallest integral value possible for c.

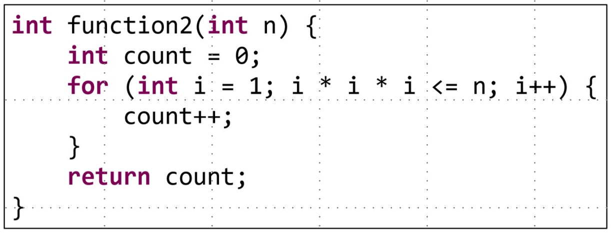
4. (5 points) Find an asymptotically tight bound for $f(n) = 3n^3 - 2n$. Prove your answer by giving values for the constants c_1, c_2, and n_0. Choose the tightest integral values possible for c_1 and c_2.

5. (5 points) Is $3n - 4 \in \Omega\left(n^2\right)$? Prove your response.

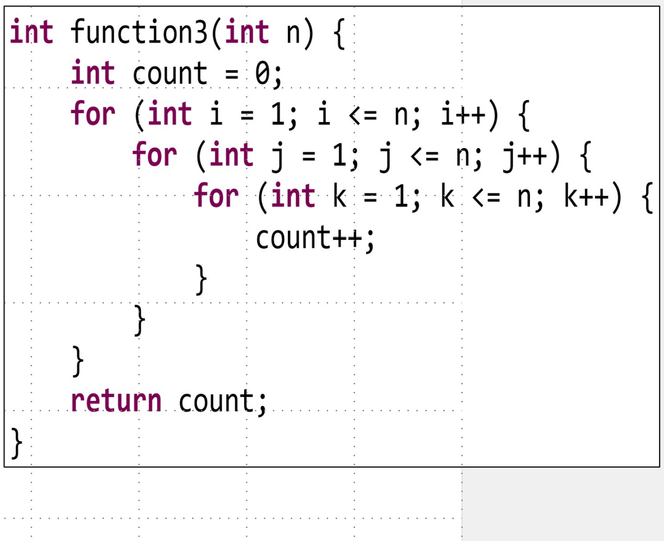
6. (10 points) Express the complexity of the following functions with the most appropriate notation



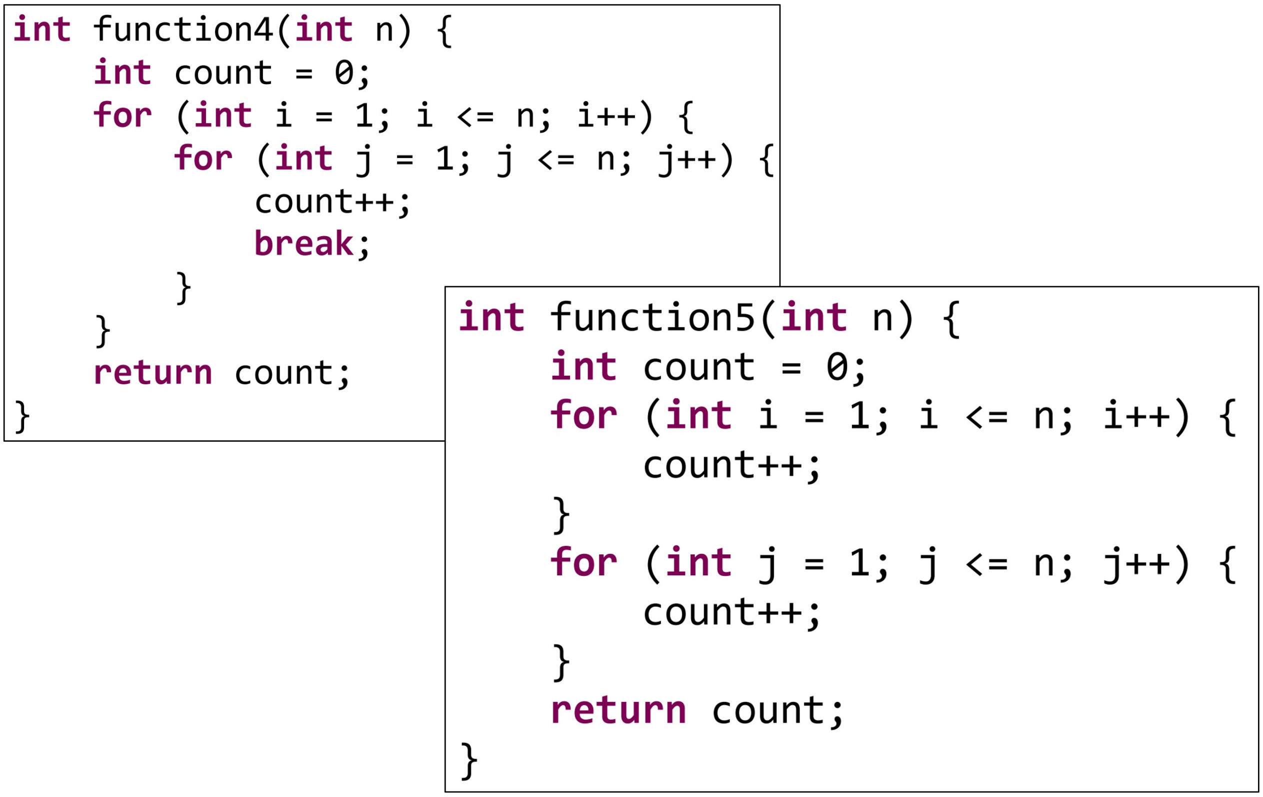
```
int function1(int n) {
    int count = 0;
    for (int i = n / 2; i <= n; i++) {
        for (int j = 1; j <= n; j *= 2) {
            count++;
        }
    }
    return count;
}
```



```
int function2(int n) {
    int count = 0;
    for (int i = 1; i * i * i <= n; i++) {
        count++;
    }
    return count;
}
```



```
int function3(int n) {
    int count = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            for (int k = 1; k <= n; k++) {
                count++;
            }
        }
    }
    return count;
}
```



```
int function4(int n) {
    int count = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            count++;
            break;
        }
    }
    return count;
}
```

```
int function5(int n) {
    int count = 0;
    for (int i = 1; i <= n; i++) {
        count++;
    }
    for (int j = 1; j <= n; j++) {
        count++;
    }
    return count;
}
```

7. (10 points) Express the runtime complexity of the following code fragments in appropriate notation (find the tightest/closest function) as functions of *m* and/or *n*? Explain your response.

a.
```
int sum = 0;
for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j+=2)
                sum += (i+j);
```

b.
```
int sum = 0;
for (int i = 1; i <= 50; i+=2)
        for (int j = 1; j <= n; j+=3)
                sum += (i+j);
```

c.
```
int sum = 0;
for (int i = 1; i <= m; i++)
        for (int j = 1; j <= n; j*=2)
                sum += (i+j);
```

d.
```
int sum1 = 0;
for (int i = 1; i <= m*m; i++)
     sum1 += i;
int sum2 = 0;
for (int j = 1; j <= m; j++)
     sum2 += j;
```