

Name: Inho Choi

ID: 1801787.

**Problem 5: Big-O**

For all of the following, determine the **total operation count function** and then the **Big-O** of the given code segments. Remember that you are mostly just counting the number of times something happens, if it helps you can plug in some numbers to get a sense of how many times certain things occur, and then generalize from that.

a.

```
for (int i = 0; i < p; i++)  
    sum++;
```

$$f(p) = 1 + p + p = 3p + 1$$

$$\Rightarrow O(p)$$

b.

```
int k = m;  
while (k > 0) {  
    sum++;  
    k--;  
}
```

$$f(m) = 1 + m + m + m = 3m + 1$$

$$\Rightarrow O(m)$$

c.

```
for (int j = 0; j < z; j++) {  
    int i = 0;  
    while (i < z) {  
        sum++;  
        i++;  
    }  
}
```

$$f(z) = 1 + z + z + z (z + z + z) \\ = 3z^2 + 3z + 1$$

$$\Rightarrow O(z^2)$$

Name: Inho Choi

ID: 1801787.

For all the following just provide the Big-O:

d.

```
int magicfact(int n) {
    int mult = 1;
    for (int i = 1; i <= n; i++)
        mult *= i;
    return mult;
}
```

Given an integer  $n$ , the loop iterates  $n$  times. $O(n)$ 

e.

```
int fact(n) {
    if (n == 0)
        return 1;
    return n * fact(n - 1);
}
```

Given an integer  $n$  we make  $n$  calls to fact  
 with each call being constant time  $\Rightarrow O(n)$

f. 

```
for (int i = 0; i < q*q; i++)
    for (int j = 0; j < i; j++)
        sum++;
```

$$f(q) = 1 + 3q^2 + 3 \frac{q^2(q^2-1)}{2}$$

$$\Rightarrow O(q^4)$$

g.

```
for (int i = 0; i < n; i++)
    for (int j = 0; j < i*i; j++)
        for (int k = 0; k < j; k++)
            sum++;
```

$$\Rightarrow O(n^5).$$

Name: Inho Choi

ID: 1801787.

h.

```
for (int i = 0; i < p; i++)  
    for (int j = 0; j < i*i; j++)  
        for (int k = 0; k < i; k++)  
            sum++;
```

$\curvearrowright = P * P$   
 $\curvearrowright = P.$

 $\Rightarrow O(P^4).$ 

i.

```
for (int i = 0; i < n; i++)  
{  
    Circ arr[n];  
    arr[i].setRadius(i);  
}
```

 $\Rightarrow O(n^2)$ 

j.

```
for (int i = 0; i < n; i++)  
{  
    int k = i;  
    while (k > 1)  
    {  
        sum++;  
        k = k / 2;  
    }  
}
```

 $\Rightarrow O(n \log_2 n).$

Name: Inho Choi.

ID: 1801787.

**Problem 2:**

Given a vector of sets of ints, `vector< set<int> > v`, assume the vector `v` has `N` total sets and that each set has an average of `Q` items.

- a. What is the Big-O of determining if the first set, `v[0]`, contains the value 7?

Getting the first set is  $O(1)$  from the vector, searching the set for 7 is  $O(\log_2(Q))$ , as given by the set contains method.

$$O(1) + O(\log_2(Q)) = O(\log_2(Q)).$$

- b. What is the Big-O of determining if any set in `v` has the value 7?

Iterating the vector will be  $O(N)$ .

Searching for the 7 in the current set will be

In total the operation will be  $O(N \log_2(Q))$ .

- c. What is the Big-O of determining the number of even values in all of `v`?

$$\Rightarrow O(N \cdot Q).$$

- d. What is the Big-O of finding the first set with a value of 7 and then counting the number of even values in that set?

$$\Rightarrow O(N / \log_2(Q) + Q)$$