CSCI 104 HW 1

Dresden Kershaw

Important information:

Definition of Big-O Notation
$f(n) = O(g(n)) \iff \lim_{n\to\infty} f(n)\frac{|f(n)|}{g(n)} < \infty$

Definition of Big-Ω Notation
$f(n) = \Omega(g(n)) \iff \lim_{n\to\infty} f(n)\frac{|f(n)|}{g(n)} > 0$

Definition of Big-Θ Notation
$f(n) = \Theta(g(n)) \iff (f(n) = O(g(n))) \wedge (f(n) = \Omega(g(n)))$

1. *See Codio submission.*
2. *See Codio submission.*
3. (a)
```
void f1(int n)
{
    int i=2;
    while(i < n){
        /* do something that takes O(1) time */
        i = i*i;
    }
}
```
$\sum_2^n f(n) = \log(n+n) - 2 = \log(2n) - 2$
$\therefore f(n) = O(\log(n))$ by the Definition of Big-O Notation
$\therefore f(n) = \Omega(\log(n))$ by the Definition of Big-Ω Notation
$\therefore f(n) = \Theta(\log(n))$ by the Defintion of Big-Θ Notation

(b)
```
void f2(int n)
{
    for(int i=1; i <= n; i++){
        if( (i % (int)sqrt(n)) == 0){
            for(int k=0; k < pow(i,3); k++) {
```

```
                /* do something that takes O(1) time */
            }
        }
    }
}
```

$\sum_1^n f(n) = (n) * \log(n^3) = O(f(n))$

(c)
```
for(int i=1; i <= n; i++){
    for(int k=1; k <= n; k++){
        if( A[k] == i){
            for(int m=1; m <= n; m=m+m){
                // do something that takes O(1) time
                // Assume the contents of the A[] array are not cha
            }
        }
    }
}
```

Outermost loop $(f(n)_0) = O(n)$ $\because$ i = i++ in for() parameters; $f(n)_0 = \Theta(n) \because \sum_1^n = n \ \forall n \in \mathbb{Z} \wedge$ by the Definition of Big-O, $-\Theta$, and $-\Omega$ notations

Next inner loop $(f(n)_1) = O(n) \because$ k = k++ in for() parameters $f(n)_1 = \Theta(n) \because \sum_1^n = n \ \forall n \in \mathbb{Z} \wedge$ by the Definition of Big-O, $-\Theta$, and $-\Omega$ notations

Innermost loop $(f(n)_2) = O(log(n)) \because$ m = 2m in for() parameters $f(n)_2 = \Theta(\log(n)) \because \sum_1^n = \log(n) \ \forall n \in \mathbb{Z} \wedge$ by the Definition of Big-O, $-\Theta$, and $-\Omega$ notations

$\therefore \sum_1^n f(n)_{all} = n * n * \log(n) = n^2 \log(n)$

(d)
```
int f (int n)
{
```

```
int *a = new int [10];
int size = 10;
for (int i = 0; i < n; i ++)
  {
    if (i == size)
      {
        int newsize = 3*size/2;
        int *b = new int [newsize];
        for (int j = 0; j < size; j ++) b[j] = a[j];
        delete [] a;
        a = b;
        size = newsize;
      }
    a[i] = i*i;
  }
}
```

The inner for() loop has a runtime of O(n), since it performs a copy operation on array indices 0 through size. This may be either a fraction of the magnitude of i, or as large as i, thus O(n). This loop also has a theoretical runtime of $\Omega(1)$, since the inner for loop may not be triggered if the array insertion index is not equal to the size variable. Thus, the inner loop can be considered to have a runtime of Θ(n) by the definitions of Big-O, -Θ, and -Ω notations.

4. See attached.