CSCI 104 – Fall 2022
Homework 1
Due Thursday 9/9, 11:59pm

Name: Daniel Dirksen

Q3 Answers

(a) Code:

```
void f1(int n)
{
    int i=2;
    while(i < n){
        /* do something that takes O(1) time */
        i = i*i;
    }
}
```

Big-Theta: i takes the values of...

$$i = 2, 2*2, 2*2*2*2, 2*2*2*2*2*2*2*2, ... = 2^(2^k)$$

So it stops when

$$2^(2^k) \geq n$$

$$2^k \geq \log_2 n$$

$$k \geq \log_2(\log_2 n)$$

Using k as the index,

$$\Theta(f1(n)) = \sum_{k=0}^{\log_2(\log_2 n)} \Theta(1)$$

$$= \Theta(\log_2(\log_2 n))$$

(b) Code:

```
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 */
            }
        }
    }
}
```

Big-Theta: When the conditional is considered, I goes from...

$$\sqrt{n}, 2\sqrt{n}, 3\sqrt{n}, ...\sqrt{n}\sqrt{n} = i\sqrt{n}$$

For int i ranging from 1 to $\sqrt{n}$. Written as a sum with k as an index:

$$\Theta(f2(n)) = \sum_{i=1}^{n} \Theta(1) + \sum_{i=1}^{\sqrt{n}} \sum_{k=0}^{(\sqrt{i})^3} \Theta(1)$$

$$= \Theta(n) + \sum_{i=1}^{\sqrt{n}} \Theta(i^{3/2})$$

$$= \Theta(n) + \sqrt{n}\Theta(n^{3/2})$$

$$= \Theta(n) + \Theta(n^{1/2}n^{3/2})$$

$$= \Theta(n) + \Theta(n^2)$$

$$= \Theta(n^2)$$

(c) Code:

```
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 changed
      }
    }
  }
}
```

In words: Search entire array of length n for each integer from 1 to n. For each hit, double an index until it hits n, doing a $\Theta(1)$ task for each n.
Run-time For Search:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\Theta(1) = \Theta(n^2)$$

Run-time For Hits (Worst case is n hits):

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\{\Theta(\log_2 n)|A[k] = i\}$$

$$= \sum_{i=1}^{n}\Theta(\log_2 n)$$

$$= \Theta(n \log_2 n)$$

Total Run-time:

$$\begin{aligned}\Theta(f(n)) &= \Theta(n^2) + \Theta(n \log_2 n)\\ &= \Theta(n^2)\end{aligned}$$

(d) Code:

```
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;
     }
}
```

In words: create a 10-element array. Start i at zero and iterate up, populating it. Every time it reaches the end of the list, migrate it to one 50 % longer.

Run-time, outer *for*:

$$\sum_{i=0}^{n} \Theta(1) = \Theta(n)$$

Each time the end is met:

$$\sum_{i=0}^{n} \Theta(1) = \Theta(n)$$

The end is met every i when:

$$i = 10, 10(1.5), 10(1.5)^2, 10(1.5)^3, ..., n = 10(1.5)^i$$

Meeting the end when...

$$10(1.5)^i = n$$
$$(1.5)^i = \frac{n}{10}$$
$$i = \log_{1.5}(\frac{n}{10})$$

Written as a sum:

$$\Theta(f(n)) = \sum_{i=0}^{\log_{1.5}(n/10)} \Theta(10(1.5)^i)$$
$$= \sum_{i=0}^{\log_2(n)} \Theta(2^i)$$
$$= \Theta(2^{\log_2(n)})$$
$$= \Theta(n)$$

Combined:

$$\Theta(f(n)) = \Theta(n) + \Theta(n)$$
$$= \Theta(n)$$