

Problem 3

Part (a)

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

$i: 2, 4, 16, 256, \dots$
 $2^2 \rightarrow \log(\log(n))$

Since the while loop's execution depends on $i < n$, and i is being squared every iteration, worst case the runtime is $\log(\log(n))$.

Part (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 */
            }
        }
    }
}
```

$O(n)$
 $O(\sqrt{n})$
 $O(n^3)$
 $O(1)$

The innermost for loop determines the worst case runtime, however the if statement before has a $O(\sqrt{n})$ runtime due to the $\text{sqrt}(n)$. Since the innermost for loop is $O(n^3)$ only executes after the if statement: $n^3 \cdot \sqrt{n} = n^{3.5} \rightarrow O(n^{3.5})$

Part (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 changed
            }
        }
    }
}
```

$O(n)$
 $O(n)$
 $O(\log(n))$
 $O(1)$

The Big-O runtime for this function is going to be $O(n^2 \cdot \log(n))$ due to the there being a nested $O(\log(n))$ for loop inside 2 $O(n)$ for loops

```

int f (int n)
{
    int *a = new int [10]; → O(1)
    int size = 10; → O(1)
    for (int i = 0; i < n; i++) → O(n)
    {
        if (i == size)
        {
            int newsz = 3*size/2; → O(1)
            int *b = new int [newsz]; → O(1)
            for (int j = 0; j < size; j++) b[j] = a[j]; → O(log3/2(n))
            delete [] a; → O(1)
            a = b;
            size = newsz; → O(1)
        }
        a[i] = i*i; → O(1)
    }
}

```

The innermost loop
 will execute less than
 n times since it is
 multiplied by $3/2$
 every loop. The runtime
 would therefore be
 $O(\log_{3/2}(n) * n)$ due to
 the $O(\log_{3/2}(n))$ for loop nested
 inside an $O(n)$ for loop