# Daniel Tadesse CSCI 104 Q3

(a)

```
i=2
while(i<n){
  i=i*i
}
```

$i = 2, 4, 16, 256 \cdots, n$ or can be writen

$i = 2, 2^{1\cdot 2}, 2^{2\cdot 2}, 2^{2\cdot 2\cdot 2} \cdots n$

iteration (K): $0 \quad 1 \quad 2 \quad 3 \cdots$

$2^{2^K} \le n$

Based off that pattern, we can see our i variable is being multiplied by itself at every iteration. This patern is denoted by $2^{2^K}$, with K being the number of iterations.

This leaves us with the equation $2^{2^K} \le n$ in order to get K by itself, we have to take the $\log_2$ of $2^{2^K}$ twice.

$K = \log(\log(n))$. There for our runtime $= \Theta(\log(\log(n)))$

b)

```
for(i->n) {
  if(i % √n == 0)
    for(K -> i^3) {
    }
}
```

our for loops runs n times, however the if statement activates $\sqrt{n}$ times

$\hookrightarrow = 1\sqrt{n},\ 2\sqrt{n},\ 3\sqrt{n},\ 4\sqrt{n}, \cdots \sqrt{n}\sqrt{n} = i$

when our function activates, we will run our second loop $(c\sqrt{n})^3$ times

$(1\sqrt{n})^3, (2\sqrt{n})^3, (3\sqrt{n})^3, (4\sqrt{n})^3 \cdots (\sqrt{n}\sqrt{n})^3 = K$

That patern can be denoted by the following summation

$$\sum_{i=1}^{\sqrt{n}} (i\sqrt{n})^3 = \sum_{i=1}^{\sqrt{n}} i^3 (\sqrt{n})^3 = (\sqrt{n})^3 \sum_{i=1}^{\sqrt{n}} i^3$$

because this is a general form of a arithmetic series, it simplifies to

$$(\sqrt{n})^3 \cdot (\sqrt{n})^4 = (n^{\frac{1}{2}})^3 \cdot (n^{\frac{1}{2}})^4 = n^{\frac{3}{2}} \cdot n^{\frac{4}{2}} = n^{\frac{7}{2}} = \Theta(n^{\frac{7}{2}})$$

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
      }
    }
  }
}
```

$\}\Theta(n^2)$ — will run $\Theta(n^2)$ because its a nested for loop

— $\log(n)$ because $m = m + m$

array A = | 1 | 2 | 3 | 4 | 5 |

if $A[k] = i$

$n = 5$ if $i$ is going from 1-5, our statement will be true 5 times, ($n$) times. once every iteration

array 2A = | 1 | 1 | 1 | 1 | 1 |

if $A[k] = i$

if $i$ = going from 1-5, our if statement is true 5 times when $i = 1$. therefor $n$ times

This means no matter what, our if Statement only activates n times. and the function inside our if statement will run log(n) times. Therefore we are left with

$\Theta(n^2) + \Theta(n \log n)$ we will chose the bigger function, therefore $\boxed{\Theta(n^2)}$

d)

```
int f (int n)
{
  int *a = new int [10];
  int size = 10;
  for (int i = 0; i < n; i ++) — θ(n)
    {
      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;                                   } θ(size)
          a = b;
          size = newsize;
        }
      a[i] = i*i;
    }
}
```

a =

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

if i=size

size = 1.5 × size

a =

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

size = 15

newsize = $15 \cdot 1.5$

Our size starts at 10 and gets multiplied by 1.5 $x$ number of times. $x$ being the amount of times i == size. This gives us $= 10(1.5)^x$

So, $10(1.5)^x \leq n$

$$1.5^x \leq \frac{n}{10}$$

$$\log_{1.5}\left(\frac{n}{10}\right) = x$$

Using our summations, we get $\sum_{i=0}^{\log_{1.5}(\frac{n}{10})} 10(1.5)^i$

$$= 10 \sum_{i=0}^{\log_{1.5}(\frac{n}{10})} 1.5^i$$

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

$$= \cancel{10} \cdot \frac{n}{\cancel{10}} = \Theta(n)$$