Part A: $i = 2^{2^k}$

runtime:

$\Theta(\log(\log(n)))$

$2^{2^k} < n$

$\log(2^{2^k}) < \log(n)$

$2^k < \log(n)$

$k < \log(\log(n))$

Part B:

$\sum_{j=0}^{\sqrt{n}} \sum_{i=1}^{i^3-1} \Theta(1) = n^3$

$\sum \Theta(1) + \sum_{j=0}^{\sqrt{n}} n^3 = \Theta(n^{7/2})$

Part C:

	m	g it	n
	1	0	
$2^{\text{iterations}}$	2	1	
	4	2	
	8	3	

finding m

$m = 2^g$

$\log(m) = g\log(2)$

$g = \log(m)$

$\log(m) = \log(n)$ since its going to n anyway

first case: when if loop is true

if loop is true n times

$\sum_{k=1}^{n} \log(n) = n\log n$

second case: when if loop is not true

$\Theta(1)$ will execute n^2 times

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

take bigger term

$\Theta(n^2)$

Part d.

inner for loop

$\sum_{1}^{Size} 1 = Size = S$

10

0 1 2 3 4 5 6 7 8 9 10

if(10 = = 10) →

3·10/2

15

6

↓

15

a = 15

S:Ze=15

if if-statement is true

runs till → 15
if (15 == 15

Geometric Series

22.5

$$\sum_{i=0}^{n} c^i = \sum \frac{c^{n+1} - 1}{c - 1} = \Theta(c^n)$$

factor is $\frac{3}{2}$

$$\sum_{i=0}^{n} \frac{3}{2}^i = \sum_{i=0}^{n} \frac{3/2^{n+1} - 1}{c - 1} = \Theta(3/2^K)$$

if statement is not true

$\Theta(1)$ will execute n times

$K = f(n)$

$i < n$

$10\left(\frac{3}{2}\right)^K = n$

$\left(\frac{3}{2}\right)^K = \frac{n}{10}$

$\log\left(\frac{3}{2}\right)^{n/10} = K$

Therefore the answer is $\Theta(n)$