Problem 3.

a) The while loop iterates over values of i through x^2, thus each iteration increases val of i by power of 2. The total num of iterations until $i > n$

k iteration	n	i
0	2	2
1	4	4
2	16	16
3	256	256
4	65536	65536

$i = 2^{(2^k)} \geq n$ $\quad \log_a b = c$

↳ $\log_2 n = 2^k$ $\quad a^c = b$

$\log_2(\log_2 n) = k$

↳ $O(\log(\log n))$ runtime.

b) The loop checks if i is a multiple of sqrt(n) which only occurs once per i.

↳ for this innerloop, the inner loop sums the pow(i, 3) for all i that is multiple of sqrt(n), thus summing all the cubes of the divisors of n.

k iterations	i
1	3
2	6
3	9
4	12

for $n = 9$
↳ $\sqrt{n} = 3$

$i = k\sqrt{n}$

↳ $\sum_{i=0}^{n} (O(1) + O(\sum_{k=0}^{i^2} O(1))$

$= \sum_{i=0}^{n} (O(1) + \sum_{i}^{n-1} \sum_{i=0}^{} O(1)$

↳ $O(n) + \sum_{K=1}^{\sqrt{n}} \sum_{i=0}^{K^3-1} O(1) = O(n) + \sum_{K=1}^{\sqrt{n}} O(n^3) = O(n) + O(n^3 \cdot \sqrt{n})$

$= O(n) + O(n^{\frac{7}{2}})$ ↳ **Runtime: $O(n^{\frac{7}{2}})$**

c) $\left(\sum_{i=1}^{n} O(1) \cdot \sum_{k=1}^{n} O(1)\right) \cdot \left(\sum^{n} O\left(\sum_{m=1}^{\log_2 n} O(1)\right)\right)$

$2^m = n$

$\log_2 n = m$

$O(n \cdot n) \cdot O(n \log n)$

↳ $\boxed{O(n^2 \log n)}$

d)

size	10	15	22	33	49	73	109
n	10	15	22	40	50	60	70
# iterations	1	2	3	4	5	5	5

$10\left(\frac{3}{2}\right)^{k-1} = n \quad \frac{n}{10} = \frac{3}{2}$

$k - 1 = \log_{\frac{3}{2}} \frac{n}{10}$

$\sum_{i=0}^{n-1} O(2) + \sum_{j=0}^{\log n} \left(\frac{3}{2}\right)^j$

↳ $O(n) + \sum_{k=0}^{\log n} k = O(\log n)$

$O(n) + \boxed{O\left(\frac{3}{2}^{\log n}\right)}$

runtime.