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

→ repeats 2^{2^k} times ; ∴ $2^{2^k} \ge n$

] $\Theta(1)$

→ $i^2 = 2^2$

$\Rightarrow \log(2^{2^k}) \ge \log n$

$= 2^k \cancel{\log(2)}^{1} \ge \log(n)$

$\log(2^k) \ge \log(\log n)$

$k \log(2) \ge \log\log n$

$k \ge \log\log n \;\therefore\; \boxed{\Theta(\log\log n)}$

↑ FINAL ANSWER

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

let $n = 12$:

$i=1, \dots, i=4, \dots, i=8, \dots, i=12$

↓ inner loop exec 4^3 ↓ inner loop exec. 8^3 times ↓ inner loop exec 12^3 times

⇒ when $n=12$, inner loop runs $4^3+8^3+12^3$ times ∴ TOTAL runs:

$(1\cdot\sqrt{12})^3+(2\sqrt{12})^3+(3\sqrt{12})$

$= \sum_{k=1}^{\sqrt{n}} \theta(k\sqrt{n})^3 \rightarrow \sum_{k=1}^{\sqrt{n}} \theta(k^3\sqrt{n}^3)$

$\hookrightarrow \sqrt{n}^3 \underbrace{\sum_{k=1}^{\sqrt{n}} \theta(k^3)}_{\text{general arith. form}} \Rightarrow \sqrt{n}^3\, \theta(\sqrt{n})^4$

$= \boxed{\theta(n^{7/2})}$

FINAL ANSWER ↑

```
(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
          }
        }
      }
    }
```

→ $i \le n$ ∴ n times

→ $k \le n$ ∴ n times

→ worst-case = true for each element in A ∴ runs k times } the same

→ $2^m \le n$ ∴ $\log(n)$ times

$$\underbrace{\sum_{i=1}^{n}}_{\text{outer for-loop}} \underbrace{\sum_{k=1}^{n}}_{\text{inner for-loop}} (\underbrace{\theta(1)}_{\text{if-stat.}}) + \sum_{k=1}^{n} \theta(\underbrace{\sum_{m=1}^{\log(n)} \theta(1)}_{\text{loop inside if stat.}})) \rightarrow \theta(n^2) + \theta(n\log n)$$

$= \boxed{\theta(n^2)}$ ↖ FINAL ANSWER

→ only repeats n times because element can't be touched twice

```
(d) 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;
        }
    }
```

$\Rightarrow$ Runtime $= \sum_{i=0}^{n} \theta(1) + \sum_{i}^{?} \sum_{j}^{size} \theta(1)$

→ $\sum_{j=0}^{size} \theta(1)$

→ size $= 10\cdot\frac{3}{2}, 10\cdot(\frac{3}{2})^2, \dots, 10\cdot(\frac{3}{2})^k$

How many times does if-stat. run?

$10\cdot(\frac{3}{2})^k < n \;\therefore\; (\frac{3}{2})^k \le n/10 \;\therefore\; k < \log_{3/2}(n/10)$

$\therefore \underbrace{\sum_{i}^{?} \sum_{j=0}^{size} \theta(1) \Rightarrow \sum_{i}^{\log_{3/2}(n/10)} \sum_{j=0}^{size} \theta(1)}_{\text{dependent summations}} \Rightarrow 10 \underbrace{\sum_{j}^{k} (\tfrac{3}{2})^{j}}_{\text{independent summation}} = \theta(\tfrac{3}{2})^k, k < \log_{3/2}(n/10)$

$\therefore \theta(\frac{3}{2})^k < \theta(\frac{3}{2})^{\log_{3/2}(n/10)}$

$\therefore \theta(\frac{3}{2})^k < \theta(\frac{n}{10}) \Rightarrow \theta(n)$

FINAL ANSWER

Runtime $= \sum_{i=0}^{n} \theta(1) + \theta(n)$

$= \theta(2n) = \boxed{\theta(n)}$