1-

```
somefunction(rows, cols) {
    for (i=1; i<=rows; i++) {
        for (j=1; j<=cols; j++) {
            print(*)
        }
        print(newline)
    }
}
```

| st/ex | freq | total |
|---|---|---|
| 2 | r+1 | 2r+2 |
| 2 | r(c+1) | 2rc+2r |
| 1 | rc | rc |
| 1 | r | r |

$= 3rc + 5r + 2 = 3n^2 + 5n + 2$

$f(n) = 3n^2 + 5n + 2$

$f(n) = O(g(n)) \rightarrow f(n) \leq c \cdot g(n), \quad n > n_0$

$3n^2 + 5n + 2 \rightarrow 3n^2 + 5n^2 + 2n^2 = 10n^2$

$c = 10, \quad n_0 = 2$

$3n^2 + 5n + 2 \leq 10n^2, \quad n \geq 2$

$24 \leq 40$

$T_W = O(n^2)$ worst-case

$f(n) = \Omega(g(n)) \rightarrow f(n) \geq c \cdot g(n), \quad n \geq n_0$

$c = 3, \quad n_0 = 2$

$3n^2 + 5n + 2 \geq 3n^2, \quad n \geq 2$

$T_B = \Omega(n^2)$ best-case

$T_A = \Theta(n^2) = O(n^2) = \Omega(n^2)$

2-

```
somefunction(a,b) {
  if(b==0)
    return 1
  answer = a
  increment = a
  for(i=1; i<b; i++) {
    for(j=1; j<a; j++) {
      answer += increment
    }
    increment = answer
  }
  return answer
}
```

| s+/ex | freq | total |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 2 | b+1 | 2b+2 |
| 2 | a(b+1) | 2ab+2a |
| 2 | ab | 2ab |
| 1 | b | b |
| 1 | 1 | 1 |

$= 2a + 3b + 4ab + 7 = 4n^2 + 5n + 7$

$f(n) = 4n^2 + 5n + 7$

$f(n) = O(g(n)) \rightarrow f(n) \leq c \cdot g(n)$ $\quad n \geq n_0$

$c = 5, \; n_0 = 7$

$4n^2 + 5n + 7 \leq 5n^2, \; n \geq 7$

$4 \cdot 7^2 + 42 \leq 4 \cdot 7^2 + 49$

$T_W = O(n^2)$

worst-case

$f(n) = 1$

$f(n) = \Omega(g(n)) \rightarrow f(n) \leq c \cdot g(n)$ $\quad n \geq n_0$

$\Omega(1)$

$T_B = \Omega(1)$

best-case

$O(n^2) \neq \Omega(1)$

3-

```
someFunction(arr[], arr_len) {
  val = 0
  for(i=0; i<arr_len/2; i++)
    val = val + arr[i]
  for(i=arr_len/2; i<arr_len; i++)
    val = val - arr[i]
  if(val >= 0)
    return 1
  else
    return -1
}
```

| stlex | freq | total |
|---|---|---|
| 1 | 1 | 1 |
| 2 | len/2+1 | len+2 |
| 3 | len/2 | 3len/2 |
| 2 | len/2+1 | len+2 |
| 3 | len/2 | 3len/2 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

$= 5len + 8 = 5n + 8$

$f(n) = 5n+8$

$f(n) = O(g(n))$ $\quad f(n) \leq c \cdot g(n)$ $\quad n \geq n_0$

$c = 6$, $n_0 = 8$

$5n + 8 \leq 6n$, $n \geq 8$

$48 \leq 48$

$T_W = O(n)$

worst-case

$f(n) = 5n+8$

$f(n) = \Omega(g(n))$ $\quad f(n) \geq c \cdot g(n)$ $\quad n \geq n_0$

$c = 4$, $n_0 = 8$

$5n + 8 \geq 4n$, $n \geq 8$

$48 \geq 32$

$T_B = \Omega(n)$

best-case

$T_A = \Theta(n) = O(n) = \Omega(n)$

4-

```
someFunction(n){
  c=0
  for(i=1 to n*n)
    for(j=1 to n)
      for(k=1 to 2*j)
        c=c+1
  return c
}
```

| st/ex | freq | total |
|---|---|---|
| 1 | 1 | 1 |
| 2 | $n^2+1$ | $2n^2+2$ |
| 2 | $n^2(n+1)$ | $2n^3+2n^2$ |
| 2 | $n^3(2j+1)$ | $4n^3j+2n^3$ |
| 2 | $n^3.2j$ | $4n^3j$ |
| 1 | 1 | 1 |

$= (8j+2).n^3 + 2n^3 + 4n^2 + 4$

$= 8n^4 + 4n^3 + 4n^2 + 4$

$f(n) = 8n^4 + 4n^3 + 4n^2 + 4$

$f(n) = O(g(n)) \rightarrow f(n) \leq c.g(n)$ , $n \geq n_0$

$c = 9$ , $n_0 = 4$

$8n^4 + 4n^3 + 4n^2 + 4 \leq 9n^4$

$8.4^4 + 4.4^3 + 4.4^2 + 4 \leq 4^4 + 8.4^4$

$T_w = O(n^4)$

worst-case

$f(n) = 8n^4 + 4n^3 + 4n^2 + 4$

$f(n) = \Omega(g(n)) \rightarrow f(n) \geq c.g(n)$ , $n \geq n_0$

$c = 4$ , $n_0 = 4$

$8n^4 + 4n^3 + 4n^2 + 4 \geq 4.n^4$

$T_B = \Omega(n^4)$

best-case

$T_A = \Theta(n^4) = O(n^4) = \Omega(n^4)$

5-

```
otherfunction (xp, yp) {
    temp = xp
    xp = yp
}   yp = temp

somefunction (arr[], arr_len) {
    for (i = 0; i < arr_len - 1; i++) {
        min_idx = i
        for (j = i+1; j < arr_len; j++) {
            if (arr[j] < arr[min_idx])
                min_idx = j
        }
        otherfunction(arr[min_idx], arr[i])
    }
}
```

$f(n) = (n^2 - n)/2$

$f(n) = O(g(n)) \rightarrow f(n) \leq c \cdot g(n) \quad , \quad n \geq n_0$

$f(n) \neq O(n^2)$

$c = 1, \quad n_0 = 0$

$(n^2 - n)/2 \leq n \quad , \quad n \geq 0$

$(n - 1)/2 \leq 1$

$n - 1 \leq 2$

$n \leq 3$

$T_W = O(n)$

worst-case

$f(n) = (n^2 - n)/2$

$f(n) = \Omega(g(n)) \rightarrow f(n) \geq c \cdot g(n) \quad , \quad n \geq n_0$

$c = 1/2, \quad n_0 = 0$

$(n^2 - n)/2 \geq n/2 \quad , \quad n \geq 0$

$n - 1 \geq 1$

$n \geq 2$

$T_B = \Omega(n)$

best-case

$T_A = \Theta(n) = O(n) = \Omega(n)$

6-

```
otherfunction(a, b){
  if (b == 0)
    return 1
  answer = a
  increment = a
  for (i = 1 to b)
    for(j = 1 to a)
      answer += increment
    increment = answer
  return answer

somefunction(arr, arr_len){
  for i = 0 to arr_len
    for j = i to arr_len
      if otherfunction(arr[i], 2) == arr[j]
        print (arr[i], arr[j])
      elif otherfunction(arr[j], 2) == arr[i]
        print (arr[j], arr[i]
}
```

$f(n) = (n^4 + 2n^3 + 5n^2 - 8n)/4$

$f(n) = O(g(n)) \rightarrow f(n) \leq c.g(n)$ , $n \geq n_0$

$c = 4$ , $n_0 = 1$

$(n^4 + 2n^3 + 5n^2 - 8n)/4 \leq 4n^4$ , $n \geq 1$

$0/4 \leq 4$

$T_w = O(n^4)$ worst-case

$f(n) = (n^4 + 2n^3 + 5n^2 - 8n)/4$

$f(n) = O(g(n)) \rightarrow f(n) \geq c.g(n)$ , $n \geq n_0$

$c = 1$ , $n_0 = 3$

$(n^4 + 2n^3 + 5n^2 - 8n)/4 \geq n^4$ , $n \geq 3$

$(3^4 + 2.3^3 + 5.3^2 - 8.3)/4 \geq 3^4$ ✓

$T_B = \Omega(n^4)$ best-case

7-

```
otherfunction (x, i) {
    s = 0
    for (j=1; j<=i; j = j*2)
        s = s + x[j]
    return s
}

somefunction (arr[], arr_len) {
    for (i = 0; i <= arr_len - 1; i++)
        A[i] = otherfunction(arr, i) / (i+1)
    return A
}
```

$of(n) = 2\log_2 n$ (otherfunc.)
$sf(n) = n\log_2 n$ (somefunc.)

$sf(n) = n\log_2 n$
$sf(n) = O(g(n))$ $\quad sf(n) \leq c.g(n)$ , $n \geq n_0$

$c = 2 \qquad n_0 = 2$

$2\log_2 2 \leq \log_2 n . 2 \qquad n \geq 2$

$2 \leq 2$

$T_w = O(\log_2 n)$

worst-case

$sf(n) = \Omega(g(n))$ $\quad sf(n) \geq c.g(n)$ , $n \geq n_0$

$c = 2 \qquad n_0 = 4$

$n\log_2 n \geq 2\log_2 n \qquad n \geq 4$

$8 \geq 4$

$T_B = \Omega(\log_2 n)$

best-case

8-

```
somefunction (n) {
    res = 0
    j = 1
    if (n < 10)
        return n + 10
    for (i = 9; i >= 1; i--)
        while (n % i == 0)
            n = n / i
            res = res + j * i
            j *= 10
    if (n > 10)
        return -1
    return res
}
```

- Not Possible to calculate!

Part 2:

1-

```
somefunction(arr[][], arr_len, px, py){
    result_index = 0
    for(i=1; i < arr_len; i++)
        dx1 = arr[i][0] - px
        dy1 = arr[i][1] - py
        dx2 = arr[result_index][0] - px
        dy2 = arr[result_index][1] - py
        if(dx1*dx1 + dy1*dy1 < dx2*dx2 + dy2*dy2)
            result_index = i
    return result_index
}
```

1
2 . (n+1)
4 . n
4 . n
4 . n
4 . n
7 . n
1 . n
1

$f(n) = 26n + 4$

$f(n) = O(g(n))$ $\quad f(n) \leq c \cdot g(n)$ $\quad n \geq n_0$

$c = 30$ , $n_0 = 1$

$26n + 4 \leq 30n \quad n \geq 1$

$30 \leq 30 \quad 1 \geq 1$

$T_W = O(n)$

worst-case

$f(n) = 26n + 4$

$f(n) = \Omega(g(n))$ $\quad f(n) \geq c \cdot g(n)$ $\quad n \geq n_0$

$c = 26$ , $n_0 = 1$

$26n + 4 \geq 26n \quad n \geq 1$

$T_B = \Omega(n)$

best-case

$T_A = \Theta(n) = O(n) = \Omega(n)$

2-
a)

```
somefunction(arr[], arr_len){
  for(i = 1; i < arr_len - 1; i++)
    if(arr[i-1] >= arr[i] and arr[i+1] >= arr[i])
      return i
  return -1
}
```

n+1 / 7n / n / 1

$f(n) = 9n + 2$

$f(n) = O(g(n))$ $\quad f(n) \leq c \cdot g(n)$ $\quad n \geq n_0$

$c = 12$, $n_0 = 1$

$9n + 2 \leq 12n$ $\quad n \geq 1$

$11 \leq 12$ $\quad 1 \geq 1$

$T_W = O(n)$ worst-case

$f(n) = \Omega(g(n))$ $\quad f(n) \geq c\, g(n)$ $\quad n \geq n_0$

$c = 9$, $n_0 = 1$

$9n + 2 \geq 9n$ $\quad n \geq 1$

$11 \geq 9$ $\quad 1 \geq 1$

$T_B = \Omega(n)$ best-case

$T_A = \Theta(n) = O(n) = \Omega(n)$

2-

b)

```
someFunction(arr[], arr_len) {
    res[] = {}                                              1
    res_len = 0                                             1
    for(i=1; i< arr_len-1; i++)                             n
        if( arr[i-1] >= arr[i] and arr[i+1] >= arr[i])      7(n-1)
            res[res_len] = i                                2(n-1)
            res_len++                                       n-1
    return res[]                                            1
```

$f(n) = 10n - 7$

$f(n) = O(g(n)) \rightarrow f(n) \leq c \cdot g(n) \quad , \quad n \geq n_0$

$c = 10 \quad , \quad n_0 = 1$

$10n - 7 \leq 10n \quad n \geq 1$

$3 \leq 10 \quad 1 \geq 1$

$T_W = O(n)$ worst-case

$f(n) = \Omega(g(n)) \rightarrow f(n) \geq c \cdot g(n) \quad , \quad n \geq n_0$

$c = 3 \quad , \quad n_0 = 1$

$10n - 7 \geq 3n \quad n \geq 1$

$3 \geq 3 \quad 1 \geq 1$

$T_B = \Omega(n)$ best-case

3-

```
someFunction3( arr[], arr_len, b) {
  for(i=0; i< arr_len; i++)
    for( j= i;  j< arr_len; j++)
      if( arr[i] + arr[j] == b)
        return true
  return false
}
```

$f(n) = 4n^2 + 2n + 5$

$f(n) = O(g(n)) \rightarrow f(n) \leq c \cdot g(n)$ , $n \geq n_0$

$c = 12$ , $n_0 = 1$

$4n^2 + 2n + 5 \leq 12n^2$ $\quad n \geq 1$

$11 \leq 12$ $\quad 1 \geq 1$

$T_W = O(n^2)$

worst-case

$f(n) = \Omega(g(n)) \rightarrow f(n) \geq c \cdot g(n)$ , $n \geq n_0$

$c = 2$ , $n_0 = 1$

$4n^2 + 2n + 5 \geq 2n^2$ $\quad n \geq 1$

$11 \geq 2$ $\quad 1 \geq 1$

$T_B = \Omega(n^2)$

best-case

$T_A = \Theta(n) = O(n) = \Omega(n)$

4-

```
someFunction4( arr[], arr_len){
  for(i = arr_len - 1; 0 < i; i--)
     success = someFunction3( arr, i, arr[i])
     if( success == false) return false
  return true
}
```

$$\sum_{i=1}^{n} (4n^2 + 2n + 6) \cdot n - 1$$

$$\sum_{i=1}^{2n-2} 1$$

$$4n^3 - 4n^2 + 2n^2 - 2n + 6n - 6$$

$$4n - 1$$

$$4n^3 - 2n^2 + 8n - 7$$

$f(n) = 4n^3 - 2n^2 + 8n - 7$

$F(n) = O(g(n))$ $\quad f(n) \leq c \cdot g(n)$ , $n \geq n_0$

$c = 4$ , $n_0 = 1$

$4n^3 - 2n^2 + 8n - 7 \leq 4n^3$ ; $n \geq 1$

$4 - 2 + 8 - 7 \leq 4$ ; $1 \geq 1$

$T_W = O(n^3)$

worst-case

$f(n) = \Omega(g(n))$ $\quad f(n) \geq c \cdot g(n)$ , $n \geq n_0$

$c = 1$ , $n_0 = 1$

$4n^3 - 2n^2 + 8n - 7 \geq n^3$ ; $n \geq 1$

$4 - 2 + 8 - 7 \geq 1$ ; $1 \geq 1$

$T_B = \Omega(n^3)$

best-case