1. Xác định độ phức tạp thuật toán cho hàm **SUM**:

Function **SUM** (int x, int y):

int sum = x + y;

1. **Độ phức tạp hằng số**
2. Độ phức tạp tuyến tính
3. Độ phức tạp bình phương
4. Các đáp án đều sai
5. Xác định độ phức tạp thuật toán cho hàm **SWAP**:

Function **SWAP** (int x, int y):

if (x < y) {

int tmp = x;

x = y;

Y = tmp;

}

1. **O(1)**
2. O(x)
3. O(y)
4. Các đáp án đều sai
5. Xác định độ phức tạp thuật toán cho hàm **PRODUCT**:

Function **PRODUCT** (int x, y):

int pro = x \* y;

1. **O(1)**
2. O(x)
3. O(x\*y)
4. Các đáp án đều sai

1. Xác định độ phức tạp thuật toán cho hàm **SUM**(n):

Function **SUM** (int n):

int s = 0;

For (int x = 0; x < n; x ++)

s = s + x;

1. O(1)
2. **O(n)**
3. O(n2)
4. Các đáp án đều sai
5. Xác định độ phức tạp thuật toán cho hàm **SUM**(n):

Function **SUM** (int n):

int s = 0;

For (int x = 0; x < n; x = x + 2)

s = s + x;

1. O(1)
2. **O(n)**
3. O(n2)
4. Các đáp án đều sai
5. Xác định độ phức tạp thuật toán cho hàm **SUM(**n**)**:

Function **SUM** (int n):

int s = 0;

For (int x = 0; x < n; x ++)

if (x mod 2 == 0)

s = s + x;

else

s = s – x;

1. O(1)
2. **O(n)**
3. O(n2)
4. Các đáp án đều sai
5. Xác định độ phức tạp thuật toán cho hàm **SUM(**m, n**)**:

Function **SUM** (int m, n):

int s = 0;

For (int x = 0; x < n; x ++)

s = s + x;

For (int y = 0; y < m; y ++)

s = s \* y;

1. O(n)
2. O(m)
3. O(mn)
4. **O(m+n)**
5. Xác định độ phức tạp thuật toán cho hàm **PRODUCT(**m, n**)**:

Function **PRODUCT** (int m, int n):

int pro = 0;

For (int x = 0; x < n; x ++)

pro = pro + x ;

For (int x = 0; x < m; x ++)

For (int y = 0; y < n; y ++)

pro = pro + x \* y;

1. O(m) + O(n)
2. O(n2)
3. **O(mn)**
4. Các đáp án đều sai
5. Xác định độ phức tạp thuật toán cho hàm **BINARY(**n**)**:

Function **BINARY** (int n):

int sum = 0;

for (int x = 1; x <n; x ++)

int sum = sum + x;

for (int x = 1; x <n; x = x \*2)

sum = sum + x;

1. **O(logn)**
2. O(n)
3. O(n/2)
4. Các đáp án đều sai
5. Độ phức tạp các thuật toán được sắp xếp từ nhỏ đến lớn. Sắp xếp nào sau đây là đúng:
6. O(1) < O(n) < O(logn) < O(n2)
7. O(1) < O(logn) < O(n) < O(n2)
8. O(1) < O(n) < O(n2) < O(logn)
9. O(logn) < O(1) < O(n) < O(n2)