Bài 1:

A, Output của thuật toán là tổng bình phương các số từ 1 đến n.

B, Basic operation của thuật toán là i*i

C, Số lần thực thi basic operation:

$$C_n = \sum_{i=1}^{n} 1 = n$$

D, Lớp hiệu năng của thuật toán là Θ(n) do $C_n = n \in \Theta(n)$

E,

//Input: A non-negative interger n

S ← $\frac{n(n+1)(2n+1)}{6}$

Return **S**

Lớp hiệu năng $\Theta(1)$

Bài 2:

A, Output của thuật toán là hiệu giữa phần tử lớn nhất và phần tử bé nhất trong mảng.

B, Basic operation của thuật toán là A[i] < minval và A[i] > maxval.

C, Số lần thực thi basic operation:

$$C_n = \sum_{i=1}^{n-1} 2 = 2(n-1) = 2n - 2 \approx 2n$$

D, Lớp hiệu năng của thuật toán là Θ(n) do $C_n = 2n \in \Theta(n)$

E, Không có do việc tìm phần tử lớn nhất và nhỏ nhất sẽ cần phải kiểm tra toàn bộ phần tử

Bài 3:

A, Output của thuật toán trả về true nếu mảng đối xứng qua đường chéo chính trả về false nếu ngược lại.

B, Basic operation của thuật toán là A[i,j] != A[j,i]

C, Số lần thực thi basic operation:

$$C_n = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} (n-1) - (i+1) + 1 = \sum_{i=0}^{n-2} n - i - 1 = \frac{n(n-1)}{2} \approx \frac{n^2}{2}$$

D, Lớp hiệu năng của thuật toán là Θ(n^2) do $C_n = \frac{n^2}{2} \in \Theta(n^2)$

E, Có thể giảm thời gian bằng cách chỉ xét các điểm ở tam giác trên

ALGORITHM EnigmaImproved(A[0..n − 1, 0..n − 1])

//Input: A matrix A[0..n − 1, 0..n − 1] of real numbers

```
for i ← 0 to n − 2 do
    for j ← i + 1 to n − 1 do
        if A[i, j] != A[j, i]:
        return false
return true
```

Dù vậy, độ phức tạp thuật toán vẫn là $\Theta(n^2)$

Bài 4:

A, Hiệu năng thời gian của thuật toán:

$$C_n = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=i}^{n} 1$$

$$= \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} n - i - 1$$

$$= \sum_{i=0}^{n-2} (-n + i + 1)^2 = \frac{1}{6}(2n^3 - 3n^2 + n) \approx \frac{n^3}{3}$$

Lớp hiệu năng của thuật toán : $C_n = \frac{n^3}{3} \in \Theta(n^3)$

B, Thuật toán trên kém hiệu quả do sử dụng 3 vòng lặp for nằm chồng lên nhau khiến cho độ phức tạp thời gian tăng lên n^3.

Thuật toán trên có tác dụng khiến cho các phần tử nằm dưới đường chéo chính của ma trận thành 0 do đó ta có thể tối ưu thuật toán như sau:

```
//input: An n x (n+1) matrix

for i← 1 to n-1:

    for j ← 0 to i-1:

        arr[i][j]=0
```