

```
[3] def DecToBi():
    n = int(input('Nhập n > 0: '))
    result = ''
    while (n > 0):
        result = str(n%2) + result
        n = n // 2
    print(result)

DecToBi()

Nhập n > 0: 8
1000
```

Hàm chuyển từ số thập phân sang nhị phân, kết quả result có kiểu dữ liệu là string, % là toán tử chia lấy dư, // là toán tử chia rồi làm tròn xuống. Ý tưởng, ta chia 2 rồi làm tròn số nhập vào cho đến khi số đó bé hơn 0 thì dừng vòng lặp, trong quá trình đó ta sẽ lần lượt cập nhật result bằng cách thêm ở bên trái kết quả của phép chia lấy dư n cho 2.

```

import numpy as np
import matplotlib.pyplot as plt

def DecToBi_and_draw():
    decimal_array = np.arange(100, 1001, 100)

    gan_list = list()
    sosanh_list = list()
    log2_list = np.log2(decimal_array)

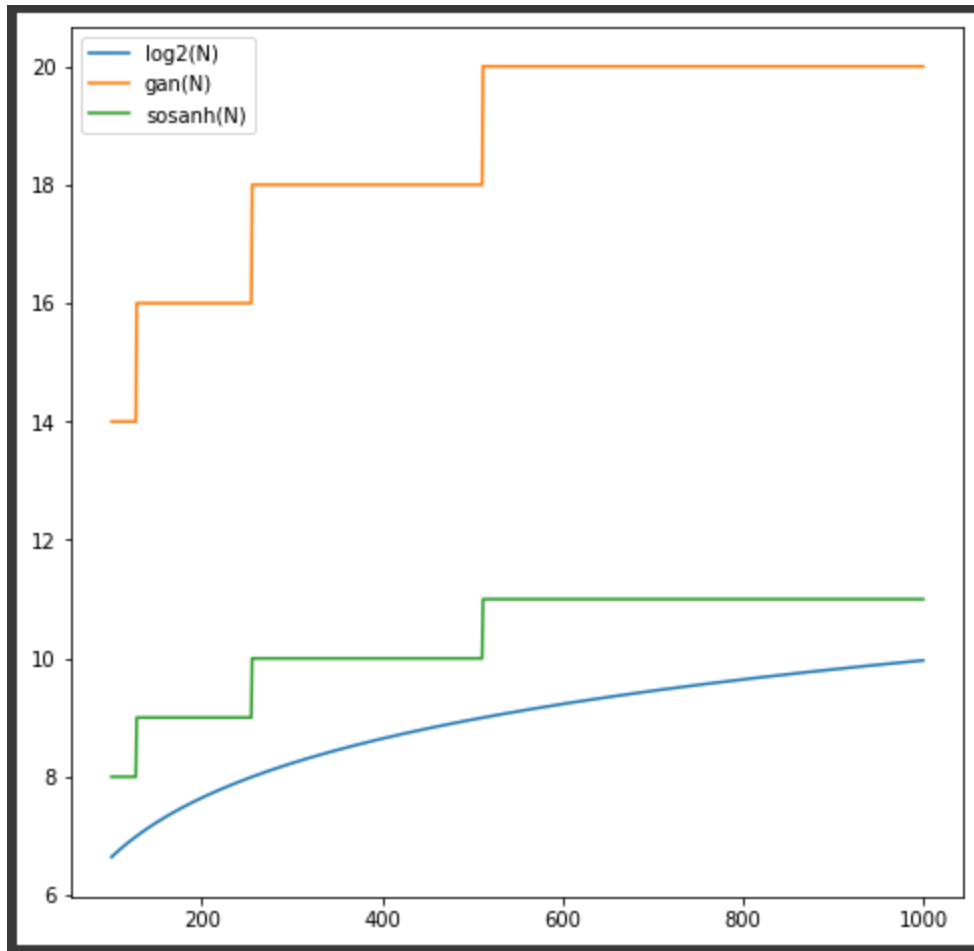
    for n in decimal_array:
        gan = 0
        sosanh = 1
        result = ''
        while (n > 0):
            result = str(n%2) + result
            n = n // 2
            gan += 2
            sosanh += 1
        gan_list.append(gan)
        sosanh_list.append(sosanh)

    plt.figure(figsize=(8, 8))
    plt.plot(decimal_array, log2_list, label = "log2(N)")
    plt.plot(decimal_array, gan_list, label = "gan(N)")
    plt.plot(decimal_array, sosanh_list, label = "sosanh(N)")
    plt.legend()
    plt.show()

DecToBi_and_draw()

```

Khá tương tự như phía trên, ở đây ta thêm vào một số biến để đếm số phép so sánh và gán, sau đó lưu vào mảng để vẽ biểu đồ. Tham số truyền vào sẽ là một mảng số thay vì chỉ một số duy nhất, ta dùng for để duyệt qua hết.



Dễ thấy,  $Gan(N)$  và  $Sosanh(N)$  có dạng giống nhau, xét về độ lớn thì do cách ta code ở trên nên  $Gan(N) = 2 * (Sosanh(N) - 1)$ .

Hơn nữa, ta có thể dễ ý thấy,  $Gan(N)$  và  $Sosanh(N)$  cũng có dạng khá giống  $\log_2 N$ ,  $N$  càng lớn thì giá trị thay đổi càng ít.