

# Phân Tích Thuật Toán

Trường Đại Học Khoa Học Tự Nhiên  
Nguyễn Thanh Bình

---

## Report lab 1

Nguyễn Quốc Bảo - 18110053

**Bài 1:** 1. Viết chương trình để biểu diễn một số thập phân N sang dạng biểu diễn nhị phân có độ phức tạp thuật toán là  $O(\log_2 N)$ .

Ta thiết lập hàm biến đổi số thập phân sang nhị phân bằng hàm **decimal -to-binary**

```
def decimal_to_binary(num):  
    count_compare = 0  
    count_assign = 0  
    Bin = 0  
    temp = num  
    i = 1  
    count_assign += 3  
    while (temp > 0):  
        count_compare += 1  
        Bin = Bin + (temp % 2)*i  
        temp = temp // 2  
        i = i*10  
        count_assign += 3  
    return Bin, count_compare, count_assign
```

Gọi số lần thực hiện vòng lặp con là  $\alpha_i$ , ta có tính

$$\alpha_i = \lfloor \log_2 i \rfloor + 1$$

Khi đó, ta tính được số

. Phép so sánh của thuật toán:  $\alpha_i + 1 = \lfloor \log_2 i \rfloor + 1$

. Phép gán của thuật toán:  $3 + 3\alpha_i = 3 + 3\lfloor \log_2 i \rfloor$

Tổng số lần thực hiện 2 Phép là:  $4\lfloor \log_2 i \rfloor + 1$

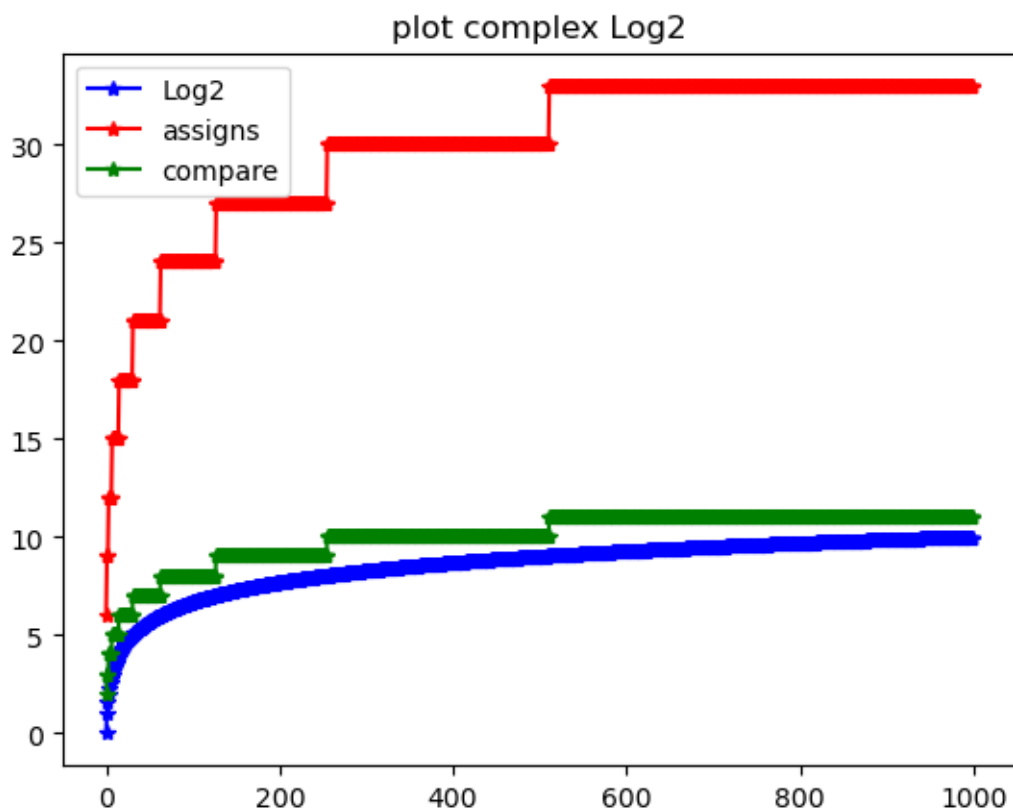
Vậy thuật toán có độ phức tạp là  $O(\log_2 n)$

```
Bài 1:  
Câu 1  
Nhập N: 28  
convert 28 to Bin: 11100  
*****
```

Và ta có kết quả như trên

2. Giả sử các em đã xây dựng thuật toán trên. Hãy viết một chương trình để đếm số phép gán và số phép so sánh mà chương trình trên đã dùng để biểu diễn một số thập phân N.

```
Câu 2
Count step assign: 6 and compare: 2 of Binary(1) = 1
Count step assign: 9 and compare: 3 of Binary(2) = 10
Count step assign: 9 and compare: 3 of Binary(3) = 11
Count step assign: 12 and compare: 4 of Binary(4) = 100
Count step assign: 12 and compare: 4 of Binary(5) = 101
Count step assign: 12 and compare: 4 of Binary(6) = 110
Count step assign: 12 and compare: 4 of Binary(7) = 111
Count step assign: 15 and compare: 5 of Binary(8) = 1000
Count step assign: 15 and compare: 5 of Binary(9) = 1001
Count step assign: 15 and compare: 5 of Binary(10) = 1010
Count step assign: 15 and compare: 5 of Binary(11) = 1011
Count step assign: 15 and compare: 5 of Binary(12) = 1100
Count step assign: 15 and compare: 5 of Binary(13) = 1101
```



**Bài 2:** Cho một mảng A gồm n số tự nhiên nằm  $[1 \dots k]$ . Hãy thiết kế thuật toán để kiểm tra xem có bao nhiêu phần tử của A nằm trong  $[a, b]$  có độ phức tạp là  $O(N + k)$  và chứng minh thuật toán của các em đưa ra có độ phức tạp như trên.

Ta xây dựng thuật toán theo như đề bài

```
def Bonus(A, a, b, k, n):
    count_assign = 0
    count_compare = 0

    Sum = 0
    counter = defaultdict(int)
    i = 1
    j = 1
    count_assign += 4
    while i < n:
        count_compare += 2
        if A[i] <= b and A[i] >= a:
            counter[A[i]] += 1
            count_assign += 1
        i += 1
    count_compare += 1
    while j < k:
        count_compare += 2
        if True:
            Sum = Sum + j
            count_assign += 1
        j += 1
        count_assign += 1
    count_compare += 1
    return count_assign, count_compare
```

Gọi số lần thực hiện vòng lặp con là  $\alpha_i$ , ta có tính

$$\alpha_1 = \sum_{i=1}^n 1 = n$$

Khi đó, ta tính được số

- . Phép so sánh của vòng lặp thứ nhất:  $2\alpha_1 + 1 = 2n + 1$
- . Phép gán của vòng lặp thứ nhất:  $2\alpha_i = 2n$

$$\alpha_2 = \sum_{j=1}^k 1 = k$$

Khi đó, ta tính được số

- . Phép so sánh của vòng lặp thứ 2:  $2\alpha_2 + 1 = 2k + 1$
- . Phép gán của vòng lặp thứ 2:  $2\alpha_i = 2k$

Số lần thực hiện:

. Phép so sánh của thuật toán:  $2(n + k) + 2$

. Phép gán của thuật toán:  $4 + 2(n + k)$

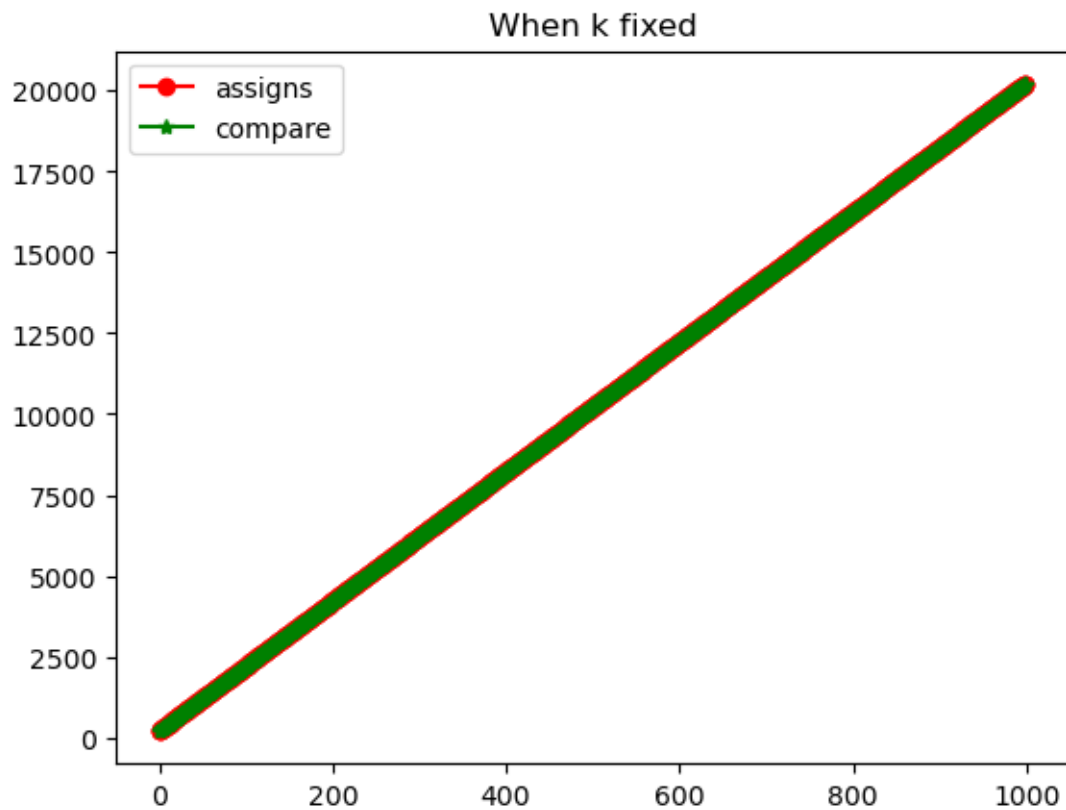
Tổng số lần thực hiện 2 Phép là:  $4(n + k) + 6$

Vậy thuật toán có độ phức tạp là  $O(n + k)$

Ta xét 2 trường hợp cần xét:

.  $N = 10, 20, 30, \dots, 10000$  (k cố định,  $k = 100$ ).

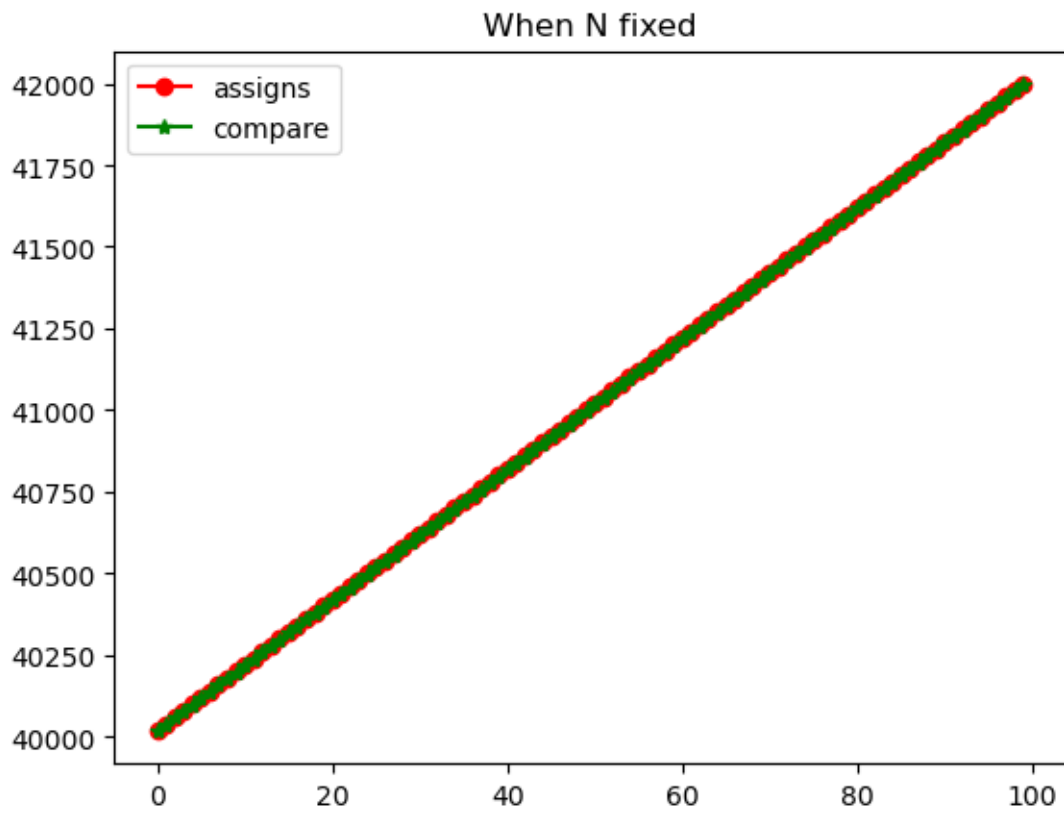
```
When k fixed 100, N: 7910
count step assign: 16020 and compare: 16018
When k fixed 100, N: 7920
count step assign: 16040 and compare: 16038
When k fixed 100, N: 7930
count step assign: 16060 and compare: 16058
When k fixed 100, N: 7940
count step assign: 16080 and compare: 16078
When k fixed 100, N: 7950
count step assign: 16100 and compare: 16098
When k fixed 100, N: 7960
count step assign: 16120 and compare: 16118
When k fixed 100, N: 7970
count step assign: 16140 and compare: 16138
When k fixed 100, N: 7980
count step assign: 16160 and compare: 16158
When k fixed 100, N: 7990
count step assign: 16180 and compare: 16178
When k fixed 100, N: 8000
count step assign: 16200 and compare: 16198
```



Với trường hợp k cố định, thì khi đó độ phức tạp của thuật toán gần bằng với  $O(N + k)$

. k = 10, 20, 30, . . . (N cố định, N = 20000).

```
When N fixed 20000, k : 850
count step assign: 41700 and compare: 41698
When N fixed 20000, k : 860
count step assign: 41720 and compare: 41718
When N fixed 20000, k : 870
count step assign: 41740 and compare: 41738
When N fixed 20000, k : 880
count step assign: 41760 and compare: 41758
When N fixed 20000, k : 890
count step assign: 41780 and compare: 41778
When N fixed 20000, k : 900
count step assign: 41800 and compare: 41798
When N fixed 20000, k : 910
count step assign: 41820 and compare: 41818
When N fixed 20000, k : 920
count step assign: 41840 and compare: 41838
```



Với trường hợp  $N$  cố định, ta cũng thấy được độ phức tạp của thuật toán cũng gần bằng với  $O(N + k)$