Bài tập về nhà phân tích độ phức tạp thuật toán đệ quy

**Bài 1:**

1. Sau mỗi lần gọi đệ quy **n giảm đi 1** nên độ phức tạp là ***O(n)***
2. Sau mỗi lần gọi đệ quy **n giảm đi một nửa** nên độ phức tạp là ***O(log2(n))***
3. Sau mỗi lần gọi đệ quy **n giảm đi một phần tư** nên độ phức tạp là ***O(log4(n))***

Trong các công thức trên, ta chỉ *gọi đệ quy một lần duy nhất*.

**Bài 2:**

1. Đoạn code sau là thuật toán***tìm kiếm nhị phân*** dùng để tìm vị trí của khóa val trong mảng b.

Return:

- Vị trí của khóa val trong mảng ***b*** [0, len(b)-1]

- Trả về -1 nếu không tồn tại khóa val trong mảng b

1. **Phần cơ sở:**

if left < right: 1 return -1

mid = (left + right) // 2

if b[mid] == val: return mid

**Phần đệ quy:**

elif b[mid] > val: return Search(val, left, mid - 1)

else: return Search(val, mid + 1, right)

1. Gọi l, r là vị trí đầu tiên và vị trí cuối cùng cần tìm kiếm

Đặt m = (l+r) // 2

**T(l, r) =**

* Nếu b[m] == val → m
* Nếu b[m] > val → T(l, m-1)
* Nếu b[m] < val → T(m+1, r)

Ta thấy rằng **không gian tìm kiếm [l, r] giảm đi một nửa** sau mỗi lần gọi đệ quy vì vậy độ phức tạp là ***O(log(L))***, với L là chiều dài của mảng cần tìm kiếm.

**Bài 3:**

1. **T(n) =**

* Nếu n == 0 → 0
* Nếu n == 1 → 2
* Ngược lại, → T(n-2) + 2

1. Thuật toán trên không tối ưu vì khi **n là số lẻ**, ta chưa tận dụng được hết thời gian in song song của 2 máy.

Xét ví dụ, với n = 3, thời gian chạy trên 2 máy sẽ là: T(1) + 2 = 2 + 2 = 4

Trong khi ta hoàn toàn có thể đạt được thời gian nhỏ nhất là 3.

Nguyên nhân là vì theo cách chia như trên **khi chỉ còn lại 1 tờ giấy** thì chỉ có một máy có thể hoạt động, chứ không thể để 2 máy hoạt động song song cùng lúc.

1. Thuật toán tốt hơn là đánh lại số thứ tự cho các tờ giấy từ 0 đến n-1. Định nghĩa ***i, j*** như sau:

Tại mỗi lượt, máy 1 sẽ **in mặt trước** của tờ giấy thứ **i** và máy 2 sẽ **in mặt sau** của tờ giấy thứ **j**.

Công thức:

**T(i, j) =**

* Nếu i == 0 and j == -1 → 2 (xảy ra khi n == 1)
* Nếu i == -1 → 0
* Ngược lại, T(i, j) = T(i-1, (j-1+n) % n) + 2

Khi đó lời giải của bài toán là T(n-1, n-2).

Nói cách khác theo thuật toán trên, tại mỗi lượt 2 máy sẽ luôn in trên 2 tờ giấy khác biệt nhau, (trừ phi ngay từ đầu chỉ có 1 tờ giấy duy nhất n == 1) do đó luôn tận dụng được thời gian chạy song song của cả 2 máy (n phút với n tờ giấy, ngoại trừ 2 phút với n == 1)