

Problem 1. Sau buổi thảo luận "Phân tích độ phức tạp của thuật toán không đệ quy", Hoàng Long và Cẩm Nguyên đã áp dụng các kiến thức thu được để phân tích một thuật toán bí mật. Chỉ biết rằng, trong trường hợp trung bình, thuật toán đó có độ phức tạp thời gian là $\theta(N \log N)$

Với mỗi câu dưới đây, hãy nhận xét nó "Đúng" hay "Sai", hay "Có thể đúng hoặc sai". Giải thích lí do.

1. Trong trường hợp xấu nhất, độ phức tạp thời gian của thuật toán là $\Omega(N^3)$
2. Trong trường hợp xấu nhất, độ phức tạp thời gian của thuật toán là $O(N)$
3. Trong trường hợp tốt nhất, độ phức tạp thời gian của thuật toán là $\Omega(N^2)$

Solution. Theo đề ta thấy độ phức tạp thuật toán trong trường hợp trung bình thuộc $\theta(N \log N)$, tức là đồng thời thuộc $O(N \log N)$ và $\Omega(N \log N)$. Trong trường hợp xấu nhất, độ phức tạp thuật toán chắc chắn thuộc $\Omega(N \log N)$ và không thể thuộc $O(g(N))$ với $g(N)$ cấp thấp hơn $N \log N$; trong trường hợp tốt nhất, độ phức tạp thuật toán chắc chắn thuộc $O(N \log N)$ và không thể thuộc $\Omega(g(N))$ với $g(N)$ cấp cao hơn $N \log N$.

1. Có thể đúng hoặc sai. Nếu độ phức tạp trong trường hợp xấu nhất thuộc $\theta(N^2)$ thì câu trên sai; nếu độ phức tạp trong trường hợp xấu nhất thuộc $\theta(N^3)$ thì câu trên đúng.
2. Sai. Theo phân tích ban đầu, vì N cấp thấp hơn $N \log N$ ($\lim_{N \log N} \frac{N}{N \log N} = 0$) nên độ phức tạp thời gian của thuật toán trong trường hợp xấu nhất không thể là $O(N)$.
3. Sai. Theo phân tích ban đầu, vì N^2 cấp cao hơn $N \log N$ ($\lim_{N \log N} \frac{N^2}{N \log N} = +\infty$) nên độ phức tạp thời gian của thuật toán trong trường hợp tốt nhất không thể thuộc $\Omega(N^2)$

□

Problem 2. Cho đoạn code Python sau, tính độ phức tạp thời gian và giải thích cách tính.

```
for i in range(n // 2, n + 1): (1)
    j = 1
    while j <= n: (2)
        k = 1
        while k <= n: (3)
            count += 1
            k = k * 2
        j = j * 2
```

Solution.

1. Bước 1: Xác định tham số thể hiện input size: $\frac{n^3}{2}$

2. Bước 2: Phép tính trọng tâm thuật toán trên: phép so sánh
3. Bước 3: Số lần phép tính trọng tâm được thực thi $= \frac{n}{2} \times \log n \times \log n = n \log^2 n$
→ Vậy độ phức tạp là: $O(n \log^2 n)$

□