

Phân Tích Đánh Giá Thuật Giải

Mục tiêu

Bài tập này nhằm mục đích rèn luyện cho sinh viên khả năng phân tích và đánh giá độ phức tạp của thuật giải thông qua các phương pháp toán học như tính toán Big-O, Small-o, Big-Ω, Big-Θ. Nắm được các kỹ thuật này, sinh viên sẽ có khả năng phân tích đánh giá được độ phức tạp của các thuật giải, từ đó biết cách xác định và lựa chọn các thuật giải tối ưu để áp dụng vào các bài toán thực tế.

Nội dung

Câu 1 Tính Big-O của các hàm số sau:

1/ $f(n) = 5n^4 + 3n^3 + 2n^2 + 4n + 1$

2/ $f(n) = 5n^2 + 3n\log n + 2n + 5$

3/ $f(n) = 20n^3 + 10n\log n + 5$

4/ $f(n) = 3\log n + 2$

5/ $f(n) = 2n+2$

6/ $f(n) = 2n + 100\log n$

7/ $f(n) = 2n$

8/ $f(n) = 10$

Chú ý: $\log n = \log_2 n$

Câu 2 Tính độ phức tạp của thuật giải sau:

```
1  sum = 0;
2  for (int i = 0; i < n; i++) {
3      System.out.println(sum);
4      sum = sum + i;
5  }
```

Câu 3 Tính độ phức tạp của thuật giải sau:

```
1  Bước 1: Gán tổng = 0. Gán i = 0.
2  Bước 2:
3      ▪ Tăng i thêm 1 đơn vị
4      ▪ Gán tổng = tổng + i
5  Bước 3: so sánh i với 10
6      ▪ Nếu i < 10, quay lại bước 2
7      ▪ Ngược lại, nếu i ≥ 10, dừng thuật toán
```

Gợi ý: số phép gán của thuật toán là bao nhiêu ? Số phép so sánh là bao nhiêu ?

Câu 4 Tính độ phức tạp của thuật toán tính tổng dãy số sau:

$$S = 1 + \frac{1}{2} + \frac{1}{6} + \dots + \frac{1}{n!}$$

Câu 5 Cho mảng A có chứa các số nguyên, có n phần tử, giải thuật sau xác định mảng A có chứa số nguyên x hay không

```
1  i = 0;  
2  while (i < n && A[i] != x)  
3      i++;
```

Tính độ phức tạp của giải thuật trên

Câu 6 Cho 3 thuật toán dưới đây, hãy xác định thuật toán có khả năng xử lý tối ưu nhất:

Thuật toán 1:

```
1  for (i=0; i<n; i++)  
2      for (j=0; j<n; j++)  
3          k++;
```

Thuật toán 2:

```
1  for (i=0; i<n; i++)  
2      k++;  
3  for (i=0; i<n; i++)  
4      for (j=0; j<n; j++)  
5          k++;
```

Thuật toán 3:

```
1  for (int i=0; i<n-1; i++)  
2      for (int j=0; j<i; j++)  
3          k += 1;
```

Câu 7 Xác định độ phức tạp của thuật toán sau:

```
1  int MaxSubSum1(const int a[], int n) {  
2      int maxSum=0;  
3      for (int i=0; i<n; i++)  
4          for (int j=i; j<n; j++) {  
5              int thisSum=0;  
6              for (int k=i; k<=j; k++)      thisSum+=a[k];  
7              if (thisSum>maxSum) maxSum=thisSum;  
8          }  
9      return maxSum;
```

10 }

Câu 8 Xác định độ phức tạp của thuật giải sau:

```
1  int MaxSubSum4(const int a[], int n) {
2      int maxSum=0, thisSum=0;
3      for (int j=0; j<n; j++) {
4          thisSum+=a[j];
5          if (thisSum>maxSum) maxSum=thisSum;
6          else if (thisSum<0) thisSum=0;
7      }
8      return maxSum;
9  }
```

Câu 9 Hãy cho biết độ phức tạp của giải thuật sau:

```
1  sum = 0;
2  for (j = 0; j < n; j++)
3      for (k = 0; k < n*n; k++)
4          sum++;
```