Lab02-Algorithm Analysis

Exercises for Algorithms by Nengjun Zhu, 2022-2023 Fall Semester.

Name: 邓腾浩 **Student ID:** 22721544 **Email:** mitdth@163.com

1. Consider the sorting algorithm shown in Alg.1, which is called BUBBLESORT.

   (a) What is the minimum number of element comparisons? When is this minimum achieved?

   (b) What is the maximum number of element comparisons? When is this maximum achieved?

   (c) Express the running time of Alg.1 in terms of the O and Ω notations.

   (d) Can the running time of the algorithm be expressed in terms of the Θ notation? Explain.

Algorithm 1: BUBBLESORT

```
input  : An array A[1···n] of n elements.
output: A[1···n] in nondecreasing order.

i ← 1; sorted ← false;
while i ≤ n − 1 and not sorted do
    sorted ← true;
    for j ← n downto i + 1 do
        if A[j] < A[j − 1] then
            interchange A[j] and A[j − 1];
            sorted ← false;
        end
    end
    i ← i + 1
end
```

解:

(a) 当数组为升序数组时，比较次数最少，最少比较次数为 $n-1$

(b) 当数组为降序数组时，比较次数最多，最多比较次数为:$(n-1)+(n-2)+...+1=\frac{n(n-1)}{2}$

(c) 上界为 $O(n^2)$，下界为 $\Omega(n)$

(d) 数组逆序度大小就是排序的交换次数，最大逆序度为 $\frac{n(n-1)}{2}$，最小逆序度为 0，因此平均情况的逆序度为 $\frac{n(n-1)}{4}$，交换次数是 n^2 级的，而比较次数一定大于等于交换次数，算法的上界又是 $O(n^2)$，所以算法的平均复杂度为 $\Theta(n^2)$

2. For Alg.2 and Alg.3 shown below, answer the following questions respectively.

   (a) Give the maximum number of times Line 6 is executed in Alg.2 when n is a power of 3.

   (b) Give the maximum number of times Line 5 is executed in Alg.3 when n is a power of 2.

   (c) What is the time complexity of both algorithms expressed in the O and Θ notations?

Algorithm 2: COUNT1

```
count ← 0;
for i ← 1 to n do
    j ← ⌊n/3⌋;
    while j ≥ 1 do
        for k ← 1 to i do
            count ← count + 1;
            if j is even then
                j ← 0;
            else
                j ← ⌊j/3⌋;
            end
        end
    end
end
```

Algorithm 3: COUNT2

```
count ← 0;
for i ← 1 to n do
    j ← ⌊n/2⌋;
    while j ≥ 1 do
        count ← count + 1;
        if j is odd then
            j ← 0;
        else
            j ← j/2;
        end
    end
end
```

解:

(a) $(log_3 n)^2 + \frac{(n+log_3 n+1)(n-log_3 n)}{2}$

(b) $nlog_2 n$

(c) 算法 2：$O(n^2)$、

算法 3：$O(nlog_2 n)$、$\Theta(n)$

3. Fill in the blanks with either true of false:

$f(n)$	$g(n)$	$f = O(g)$	$f = \Omega(g)$	$f = \Theta(g)$
$2n^3 + 3n$	$100n^2 + 2n + 100$	false	true	false
$50n + \text{log}n$	$10n + \text{loglog}n$	false	false	true
$50n\text{log}n$	$10n\text{loglog}n$	false	true	false
$\text{log}n$	$\text{log}^2 n$	true	false	false
$n!$	5^n	false	true	false

4. Use the $\prec$ relation to order the following functions by growth rate:

$$n^{1/100}, \sqrt{n}, \text{log}n^{100}, n\text{log}n, 5, \text{loglog}n, \text{log}^2 n, (\sqrt{n})^n, (1/2)^n, 2^{n^2}, n!$$

解:

$$(1/2)^n \prec 5 \prec \text{loglog}n \prec \text{log}n^{100} \prec \text{log}^2 n \prec n^{1/100} \prec \sqrt{n} \prec n\text{log}n \prec 2^{n^2} \prec (\sqrt{n})^n \prec n!$$