经 56 王思萍 2015012527

Homework 2

1. According to the L'Hospital Rule, we have:

$$\lim_{N\to\infty}\frac{\frac{2}{N}}{37}=0$$

$$\lim_{N\to\infty}\frac{37}{N^{\frac{1}{2}}}=0$$

$$\lim_{N\to\infty}\frac{N^{\frac{1}{2}}}{N}=0$$

$$\lim_{N\to\infty}\frac{N}{N\log\log N}=\lim_{N\to\infty}\frac{1}{\log\log N}=0$$

$$\lim_{N\to\infty}\frac{N\log\log N}{N\log(N^2)}=\lim_{N\to\infty}\frac{\log\log N}{\log(N)^2}=\lim_{N\to\infty}\frac{N^{-1}(\log N)^{-1}}{2N^{-1}}=0$$

$$\lim_{N\to\infty}\frac{N\log(N^2)}{N\log N}=\lim_{N\to\infty}\frac{2\log N}{\log N}=2$$

$$\lim_{N\to\infty}\frac{N\log N}{N(\log N)^2}=\lim_{N\to\infty}\frac{\log N}{(\log N)^2}=\lim_{N\to\infty}\frac{N^{-1}}{2\,N^{-1}\log N}=0$$

$$\lim_{N\to\infty}\frac{N(\log N)^2}{N^{1.5}}=\lim_{N\to\infty}\frac{(\log N)^2}{N^{\frac{1}{2}}}=\lim_{N\to\infty}\frac{2\,N^{-1}\log N}{\frac{1}{2}N^{-\frac{1}{2}}}=\lim_{N\to\infty}\frac{4\log N}{N^{\frac{1}{2}}}=\lim_{N\to\infty}\frac{8N^{-1}}{N^{-\frac{1}{2}}}=0$$

$$\lim_{N\to\infty}\frac{N^{1.5}}{N^2}=0$$

$$\lim_{N\to\infty}\frac{N^2}{N^2\log N}=0$$

$$\lim_{N\to\infty}\frac{N^2\log N}{N^3}=\lim_{N\to\infty}\frac{\log N}{N}=0$$

$$\lim_{N\to\infty}\frac{N^3}{2^{\frac{N}{2}}}=0$$

$$\lim_{N\to\infty}\frac{2^{\frac{N}{2}}}{2^N}=0$$

∴ The rank of the growth rate, precisely, from the lowest to the highest, would be:

$\frac{2}{N}, 37, N^{\frac{1}{2}}, N, N\log\log N, N\log(N^2) = N\log N, N(\log N)^2, N^{1.5}, N^2, N^2\log N, N^3, 2^{\frac{N}{2}}, 2^N.$

Among them the growth rate of $N\log(N^2)$ is equal to that of $N\log N$. While by convention, though less precisely, we simply regard that $\{N\log\log N, N\log(N^2), N\log N, N(\log N)^2\}$ all have the growth rates that can be approximated to that of $N^{1.5}$.

2. (a)

(1)
```
/*1*/    Sum = 0;
/*2*/    for(i = 0; i < N; i++)
/*3*/        Sum++;
```

Analysis:

- Line 1 counts for 1 unit.
- Line 3 counts for 2 units per time executed (1 +, 1 =) and executed N times, so totally 2N units.
- Line 2 has hidden costs, i.e., 1 assignment, N+1 units for testing, and N units for incrementing, so totally counts for 2N + 2 units.
- To sum up, 4N + 3 units. O(N).

(2)
```
/*1*/    Sum = 0;
/*2*/    for(i = 0; i < N; i++)
/*3*/        for(j = 0; j < N; j++)
/*4*/            Sum ++;
```

Analysis:

- Line 1 counts for 1 unit.
- Line 2 counts for 2N + 2 units, according to the previous analysis.
- Line 3 counts for 2N + 2 units per time in the i-loop and executed N times, so totally $2N^2 + 2N$ units.
- Line 4 counts for 2 units per time and executed N^2 times, so totally $2N^2$ units.
- To sum up, $4N^2 + 4N + 3$ units. $O(N^2)$.

(3)
```
/*1*/    Sum = 0;
/*2*/    for(i = 0; i < N; i++)
/*3*/        for(j = 0; j < N * N; j++)
/*4*/            Sum ++;
```

Analysis:

- Line 1 counts for 1 unit.
- Line 2 counts for 2N + 2 units, according to the previous analysis.
- Line 3 counts for $2N^2 + 2$ units per time in the i-loop and executed N times, so totally $2N^3 + 2N$ units.
- Line 4 counts for 2 units per time and executed N^3 times, so totally $2N^3$ units.
- To sum up, $4N^3 + 4N + 3$ units. $O(N^3)$.

(4)
```
/*1*/    Sum = 0;
/*2*/    for(i = 0; i < N; i++)
/*3*/        for(j = 0; j < i; j++)
/*4*/            Sum ++;
```

Analysis:

- Line 1 counts for 1 unit, line 2 counts for 2N + 2 units.
- Line 3 counts for 2i + 2 units per time in the i-loop, where n iterations are executed, so $\sum_{i=0}^{N-1}(2i+2) = N^2 + N$ units in total.
- Line 4 counts for 2 units per time in the j-loop, where i iterations are executed per time in the i-loop, so $\sum_{i=0}^{N-1} 2i = N^2 - N$ units in total.
- To sum up, $2N^2 + 2N + 3$ units. $O(N^2)$.

(5)
```
/*1*/    Sum = 0;
/*2*/    for(i = 0; i < N; i++)
/*3*/        for(j = 0; j < i * i; j++)
/*4*/            for(k = 0; k < j; k++)
/*5*/                Sum ++;
```

Analysis:

- Line 1 counts for 1 unit, line 2 counts for 2N + 2 units.
- Line 3 counts for $2i^2 + 2$ units per time in the i-loop, where N iterations are executed, so $\sum_{i=0}^{N-1}(2i^2+2) = \sum_{i=0}^{N-1} 2i^2 + 2N$ units in total.
- Line 4 counts for 2j + 2 units per time in the j-loop, where i^2 iterations are executed per time in the i-loop, so $\sum_{i=0}^{N-1}\sum_{j=0}^{i^2-1}(2j+2)$ units in total.
- Line 5 counts for 2 units per time in the k-loop, where j iterations are executed per time in the j-loop, so $\sum_{i=0}^{N-1}\sum_{j=0}^{i^2-1}\sum_{k=0}^{j-1} 2 = \sum_{i=0}^{N-1}\sum_{j=0}^{i^2-1} 2j$ units in total.
- Totally $\sum_{i=0}^{N-1}\sum_{j=0}^{i^2-1} 4j + \sum_{i=0}^{N-1} 4i^2 + 4N + 3$ units. $O(N^5)$.

(6)
```
/*1*/    Sum = 0;
/*2*/    for(i = 0; i < N; i++)
/*3*/        for(j = 0; j < i * i; j++)
/*4*/            if(j % i == 0)
/*5*/                for(k = 0; k < j; k++)
/*6*/                    Sum ++;
```

Analysis:

- Line 1 counts for 1 unit, line 2 counts for 2N + 2 units, and line 3 counts for $\sum_{i=0}^{N-1} 2i^2 + 2N$ units, according to the previous analysis.
- Line 4 counts for 2 units per time in the j-loop, where i^2 iterations are executed per time in the i-loop, so $\sum_{i=0}^{N-1} 2i^2$ units in total.
- Line 5 counts for 2j + 2 units per time for each $j = t \cdot i$, so $\sum_{i=0}^{N-1}\sum_{t=0}^{i-1}(2t \cdot i + 2) = \sum_{i=0}^{N-1}\sum_{t=0}^{i-1} 2t \cdot i + \sum_{i=0}^{N-1} 2\,i$ units in total.
- Line 6 counts for 2 units per time in the k-loop, where $j = t \cdot i$ iterations are executed, so $\sum_{i=0}^{N-1}\sum_{t=0}^{i-1} 2(t \cdot i)$ units in total.
- Totally $\sum_{i=0}^{N-1}\sum_{t=0}^{i-1} 4t \cdot i + \sum_{i=0}^{N-1} 4\,i^2 + \sum_{i=0}^{N-1} 2\,i + 4N + 3$ units. $O(N^4)$.

2. (b) The following displays, in each of which the first value of N is selected such that the running time is approximately 100 ms and the sequent values are multiples of the head, are print-screens of *terminal* where the fragments are executed.

(1)

```
N = 50000000
Running time = 0.109240 s

N = 100000000
Running time = 0.206370 s

N = 150000000
Running time = 0.308376 s

N = 200000000
Running time = 0.394935 s

N = 250000000
Running time = 0.505111 s
```

(2)

```
N = 6900
Running time = 0.100090 s

N = 13800
Running time = 0.398776 s

N = 20700
Running time = 0.895167 s

N = 27600
Running time = 1.622538 s

N = 34500
Running time = 2.529074 s
```

(3)

```
N = 341
Running time = 0.100726 s

N = 682
Running time = 0.741670 s

N = 1023
Running time = 2.539146 s

N = 1364
Running time = 5.963904 s

N = 1705
Running time = 11.714345 s
```

(4)

```
N = 10000
Running time = 0.103985 s

N = 20000
Running time = 0.411605 s

N = 30000
Running time = 0.902075 s

N = 40000
Running time = 1.592202 s

N = 50000
Running time = 2.478773 s
```

(5)

```
N = 34
Running time = 0.010801 s

N = 68
Running time = 0.276850 s

N = 102
Running time = 2.144312 s

N = 136
Running time = 9.106790 s

N = 170
Running time = 27.677937 s
```

(6)

```
N = 139
Running time = 0.103106 s

N = 278
Running time = 1.531667 s

N = 417
Running time = 7.621815 s

N = 556
Running time = 24.030566 s

N = 695
Running time = 61.464090 s
```

2. (c) It can be seen from above that the growth of the running time in each of the six fragments is quite similar to the growth of the correlated function of N computed in (a). Take, for example, fragment (1). With a complexity of O(N), the running time grows (approximately) linearly as the initial value of N being multiplied by 2, 3, 4 and 5 consecutively. Similarly, the running time of fragment (2) and (4), with a complexity of O(N^2), grow (approximately) quadratically in reaction to the linear growth of N. Same as the other fragments. Therefore, it can be concluded that the Big-Oh analysis in (a) is correct.