Lab01-Algorithm Analysis

CS2308-Algorithm and Complexity, Xiaofeng Gao, Spring 2023.

* Contact TA Jiale Zhang for any questions. Include both your .pdf and .tex files in the uploaded .rar or .zip file.

* Name: 张祥东 Student ID: 521030910206 Email: zhangxiangdong@sjtu.edu.cn

1. (Minimal Counterexample Principle) Prove by Minimal Counterexample Principle that any integer $n \geqslant 13$ can be written as $7x + 3y$, where x and y are non-negative integers.

Proof. We know for sure that

$$13 = 7 * 1 + 3 * 2, 14 = 7 * 2, 15 = 3 * 5, 16 = 7 * 1 + 3 * 3, \ldots, 20 = 7 * 2 + 3 * 2$$

And now we assume there exists a number **m** which is the minimum number that cannot be written as 7x + 3y.

And we know

$$m \geq 20$$

so

$$m - 7 \geq 13$$

Because **m** is the minmum number that cannot be written as 7x+3y,we know $m - 7$ can be written as $7x + 3y$ through our assumption.

Let

$$m - 7 = 7x + 3y$$

So we have

$$m = 7(x + 1) + 3y$$

which is contrary to our assumption that **m** cannot be written as the form of $7x + 3y$,so we prove that any integer $n \geq 13$ can be written as 7x + 3y.

Thus,the proof is complete.

□

2. (Complexity Class) Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{14}$ of the functions $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{13} = \Omega(g_{14})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$, with "$\succ$" and "$\equiv$". e.g. $n^2 \succ n \equiv 2^{1+\log n}$. ($\log n$ means $\log_2 n$)

$$\begin{array}{ccccc} 2^n & \log n & n^2 & n & \log\log n \\ n! & n^2 \cdot 2^n & \log(n!) & n\log n & n^{\frac{1}{\log n}} \end{array}$$

Solution. sequence:

$n! \succ n^2 \cdot 2^n \succ 2^n \succ n^2 \succ n\log n \equiv \log(n!) \succ n \succ \log n \succ \log\log n \succ n^{\frac{1}{\log n}}$

We know when $\lim_{n\to\infty} \frac{g(n)}{f(n)} = +\infty$,here comes $g(n) \succ f(n)$

I claim here that I consider the sequence as a function first and after using the L'Hospital's rule I substitute the value n into this function to get the result.And in the following prove,to simpify my prove process,I use L'Hospital's rule to the limit of sequence directly sometimes.

First,let us prove $n! \succ n^2 \cdot 2^n$

We assume that the $\lim_{n\to\infty} \frac{n!}{n^2 \cdot 2^n}$ exists and it equals to A.

And we can scale the equation down to the following form:

$$\lim_{n\to\infty} \frac{n!}{n^2 \cdot 2^n} > \lim_{n\to\infty} \frac{n(n-1)}{n^2} \lim_{n\to\infty} (\frac{2*1}{2*2*2*2} * (\frac{3}{2})^{n-4}) = +\infty$$

So it's counter to what we've assumed,thus the contrary is right.

Then we get

$$\lim_{n\to\infty} \frac{n!}{n^2 \cdot 2^n} = +\infty$$

So here comes the relation which is true

$$n! \succ n^2 \cdot 2^n$$

Second,let us prove $n^2 \cdot 2^n \succ 2^n$

It's obvious that

$$\lim_{n\to\infty} \frac{n^2 \cdot 2^n}{2^n} = \lim_{n\to\infty} n^2 = +\infty$$

So here comes the relation which is true

$$n^2 \cdot 2^n \succ 2^n$$

Third,let us prove $2^n \succ n^2$

Using L' Hospital' s rule,we obtain

$$\lim_{n\to\infty} \frac{2^n}{n^2} = \lim_{n\to\infty} \frac{2^n \ln^2 2}{2} = +\infty$$

So here comes the relation which is true

$$2^n \succ n^2$$

Fourth,let us prove $n^2 \succ n \log n$

We know that

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

Using L'Hospital's rule,we get

$$\lim_{n\to\infty} \frac{n}{\log n} = \lim_{n\to\infty} \frac{1}{\frac{1}{n \ln 2}} = \lim_{n\to\infty} n \ln 2 = +\infty$$

So here comes the relation which is true

$$n^2 \succ n \log n$$

Fifth,let us prove $n \log n \equiv \log(n!)$

For $\log(n!)$,there is

$$\int_m^{m+1} \log x dx < \log(m+1) < \int_{m+1}^{m+2} \log x dx$$

we know

$$\int_1^n \log x dx < \log(n!) < \int_2^{n+1} \log x dx$$

Thus,here comes

$$\frac{n \log n}{\int_2^{n+1} \log x dx} < \frac{n \log n}{\log(n!)} < \frac{n \log n}{\int_1^n \log x dx}$$

After simplifying

$$\frac{n \log n \ln 2}{(n+1)\ln(n+1) - 2\ln 2 - n + 1} < \frac{n \log n}{\log(n!)} < \frac{n \log n \ln 2}{n \ln n - n + 1}$$

Based on limit operation theorem,we know

$$\lim_{n\to\infty} \frac{n \log n \ln 2}{(n+1)\ln(n+1) - 2\ln 2 - n + 1} = \lim_{n\to\infty} \frac{1}{\frac{(n+1)\ln(n+1) - 2\ln 2 - n + 1}{n \log n \ln 2}} = \frac{1}{1-0-0} = 1$$

And also

$$\lim_{n\to\infty} \frac{n \log n \ln 2}{n \ln n - n + 1} = 1$$

Using squeeze theorem,we know

$$\lim_{n\to\infty} \frac{n \log n}{\log(n!)} = 1$$

So here comes the relation which is true

$$n \log n \ \equiv \ \log(n!)$$

Sixth,let us prove $\log(n!) \ \succ \ n$

Just the same as the prove in the fifth step,we know

$$\int_1^n \log x dx < \log(n!) < \int_2^{n+1} \log x dx$$

So we know

$$\frac{\int_1^n \log x dx}{n} < \frac{\log(n!)}{n}$$

And we know

$$\frac{\int_1^n \log x dx}{n} = \frac{n \ln n - n + 1}{n \ln 2}$$

And

$$\lim_{n\to\infty} \frac{n \ln n - n + 1}{n \ln 2} = +\infty$$

So

$$\lim_{n\to\infty} \frac{\log(n!)}{n} \geq \lim_{n\to\infty} \frac{n \ln n - n + 1}{n \ln 2} = +\infty$$

So here comes the relation which is true

$$\log(n!) \ \succ \ n$$

Seventh,let us prove $n \ \succ \ \log n$

Using L'Hospital's rule,we know

$$\lim_{n\to\infty} \frac{n}{\log n} = \lim_{n\to\infty} n \ln 2 = +\infty$$

So here comes the relation which is true

$$n \succ \log n$$

Eighth,let us prove $\log n \succ \log\log n$

We know that

$$\lim_{n\to\infty} \frac{\log n}{\log\log n} = \lim_{n\to\infty} \frac{\ln n}{\ln \log n}$$

By L'Hospital's rule,we know

$$\lim_{n\to\infty} \frac{\ln n}{\ln\log n} = \lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{n\log n \ln 2}} = \lim_{n\to\infty} \log n \ln 2 = +\infty$$

So here comes the relation which is true

$$\log n \succ \log\log n$$

Ninth,let us prove $\log\log n \succ n^{\frac{1}{\log n}}$

We assume that the $\lim_{n\to\infty} \frac{\log\log n}{n^{\frac{1}{\log n}}} = A$ $(A > 0)$ We know

$$\lim_{n\to\infty} n^{\frac{1}{\log n}} = \lim_{n\to\infty} e^{\frac{\ln 2}{\ln n}\ln n} = e^{\ln 2} = B$$

$$\lim_{n\to\infty} \log\log n = +\infty = C$$

Based on limit operation rules,we know $A = B * C$,but C is infinity and both A and B is a limit number,so this is contrary to our assumption. We get

$$\lim_{n\to\infty} \frac{\log\log n}{n^{\frac{1}{\log n}}} = +\infty$$

So here comes the relation which is true

$$\log\log n \succ n^{\frac{1}{\log n}}$$

Above all,the partial order

$$n! \succ n^2 \cdot 2^n \succ 2^n \succ n^2 \succ n\log n \equiv \log(n!) \succ n \succ \log n \succ \log\log n \succ n^{\frac{1}{\log n}}$$

is true.

Thus, the proof is complete.

□

3. (Time Complexity Analysis) Here are the pseudo-codes of improved Count (Alg. 1) and QuickSort (Alg. 2).

Algorithm 1: Count

Input: A positive integer n
Output: No. of times Step 6 is executed.

```
1  count ← 0;
2  for i ← 1 to n do
3      j ← ⌊n/2⌋;
4      while j ≥ 1 do
5          for k ← 1 to i do
6              count ← count + 1;
7          if j is odd then
8              j ← 0;
9          else
10             j ← j/2;
```

Algorithm 2: QuickSort

Input: An array $A[1, \cdots, n]$
Output: A sorted nondecreasingly

```
i ← 1; pivot ← A[n];
for j ← 1 to n − 1 do
    if A[j] < pivot then
        swap A[i] and A[j];
        i ← i + 1;
swap A[i] and A[n];
if i > 1 then
    QuickSort(A[1, ··· , i − 1]);
if i < n then
    QuickSort(A[i + 1, ··· , n]);
```

(a) Analyze the **worst case** time complexity of Alg. 1.

For the worst case,we assume that the

$$n = 2^d - 1$$

So

$$j = 2^{d-1} - 1$$

j is a odd number and when doing operation $j \leftarrow j/2$,the j is always odd.

The worst case for

$$j \leftarrow \lfloor n/2 \rfloor, doing\ j \leftarrow j/2\ or\ j \leftarrow 0$$

is executing d - 1 times and $j \leftarrow j/2$ every time.

So we konw the worst case time complexity for Alg.1 is

$$\sum_{i=1}^{n} \sum_{m=1}^{d-1} \sum_{k=1}^{i} 1 = \frac{n(n+1)}{2}(d-1)$$

$$d - 1 = \log(n+1) - 1$$

Thus the worst case time complexity for Alg.1 can be written as

$$\frac{n(n+1)}{2}(\log(n+1) - 1)$$

It's

$$O(n^2 \log n)$$

(b) Analyze the **average** time complexity of the QuickSort in Alg. 2.

We use T(n) to symbolize the average time needed to sort an array of size n by QuickSort. There is

$$T(n) = n + X$$

X is an unknown variable.

For an array of size n,we know there are $n-1$ kinds of partitions to divide it into two sets which contain

$$(1, n-1), (2, n-2), \dots, (n-1, 1)$$

And from the average perspective,the possibility for these situations is equal to one another.

And we know for QuickSort,one operation means divide an array into two parts. In the meantinme,it's noteworthy that the **middle element** should be noticed,which means for an array of size n,we can divide it into n forms which contain

$$(0, 1, n-1), (1, 1, n-2), (2, 1, n-3), \dots, (n-2, 1, 1), (n-1, 1, 0)$$

So here comes

$$X = \frac{1}{n}((T(0)+T(n-1))+(T(1)+T(n-2))+\dots+(T(n-2)+T(1)))+(T(0)+T(n-1))$$

$T(0) = 0$

After simplifying

$$X = \frac{2}{n}(T(1) + T(2) + \dots + T(n-1))$$

Take

$$S(n) = T(n) + T(n-1) + \dots + T(1)$$

So

$$T(n) = n + \frac{2}{n}S(n-1) \qquad (1)$$

$$T(n-1) = n - 1 + \frac{2}{n-1}S(n-2) \qquad (2)$$

We perform the following operations

$$(n)(1) - (n-1)(2)$$

we obtain

$$nT(n) = (n+1)T(n-1) + 2n - 1$$

namely

$$T(n) = \frac{n+1}{n}T(n-1) + 2 - \frac{1}{n}$$

after expanding

$$\begin{aligned}
T(n) &= \frac{n+1}{n}(\frac{n}{n-1}T(n-2) + 2 - \frac{1}{n-1}) + 2 - \frac{1}{n} = \dots \\
&= 2 + 2\frac{n+1}{n} + 2\frac{n+1}{n}\frac{n}{n-1} + \dots + 2\frac{n+1}{n}\frac{n}{n-1}\dots\frac{4}{3} + \frac{n+1}{n}\dots\frac{3}{2}T(1) \\
&\quad - (n+1)(\frac{1}{n(n+1)} + \frac{1}{(n-1)n} + \dots + \frac{1}{2*3}) \\
&= 2 + 2\frac{n+1}{n} + 2\frac{n+1}{n-1} + \dots + 2\frac{n+1}{3} - \frac{n+1}{2}T(1) - (n+1)(\frac{1}{2} - \frac{1}{n+1}) \\
&= 2(n+1)(\frac{1}{n+1} + \frac{1}{n} + \dots + \frac{1}{3}) + 2\frac{n+1}{3}T(1) - \frac{n+1}{2} + 1 \\
&= 2(n+1)(1 + \frac{1}{2} + \dots + \frac{1}{n} + \frac{1}{n+1}) + 2\frac{n+1}{3}T(1) - 3(n+1) - \frac{n+1}{2} + 1
\end{aligned}$$

We know $T(1) = 0$,so we get the average time complexity

$$T(n) = 2(n+1)\sum_{i=1}^{n+1}\frac{1}{i} - 3(n+1) - \frac{n+1}{2} + 1$$

When n is big enough,from Euler's formula, we know

$$\sum_{i=1}^{n+1}\frac{1}{i} = \ln(n+1) + C$$

C is a Euler constant,there is

$$C \approx 0.5772$$

Under such circumstance,the average time complexity is

$$T(n) = 2(n+1)\ln(n+1) + (2C - \frac{7}{2})(n+1) + 1$$

$$T(n) = 2(n+1)\ln(n+1) - 2.3456n - 1.3456$$

It's

$$O(n \log n)$$