Algorithm Hw2

Group 3

Q1. **Problem 3-2: *Relative asymptotic growths*** Indicate, for each pair of expressions (*A*, *B*) in the table below, whether *A* is *O, o, Ω, w,* or *Θ* of *B*. Assume that *k* ≥ 1, *ε* > 0, and *c* >1 are constants. Your answer should be in the form of the table with “yes” or “no” written in each box, and **you should justify your answer**.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | O | o | *Ω,* | *w* | *Θ* |
| a. |  |  | YES | YES | NO | NO | NO |
| b. |  |  | YES | YES | NO | NO | NO |
| c. |  |  | NO | NO | NO | NO | NO |
| d. |  |  | NO | NO | YES | YES | NO |
| e. |  |  | YES | NO | YES | NO | YES |
| f. |  |  | YES | NO | YES | NO | YES |

1. ,

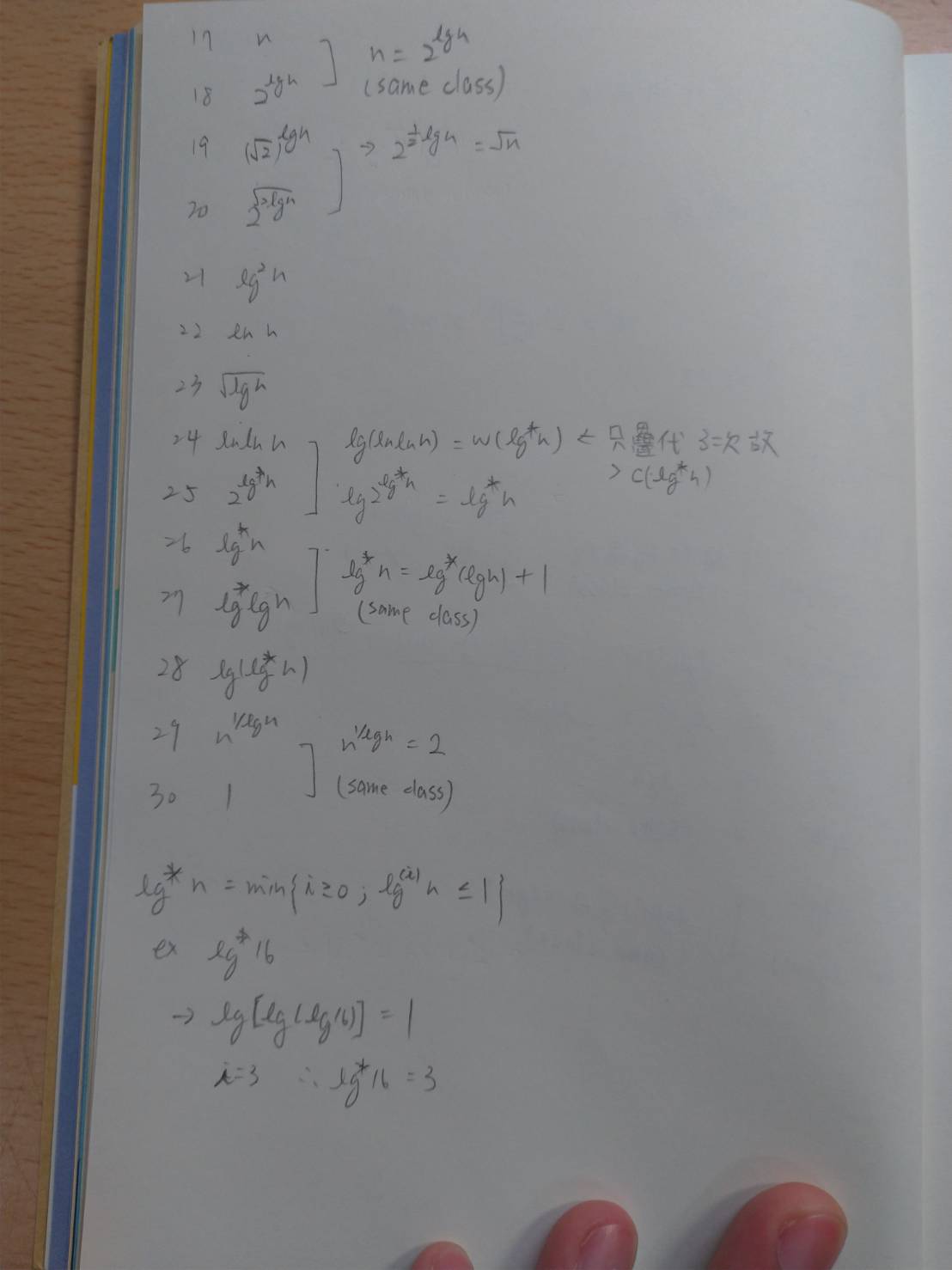
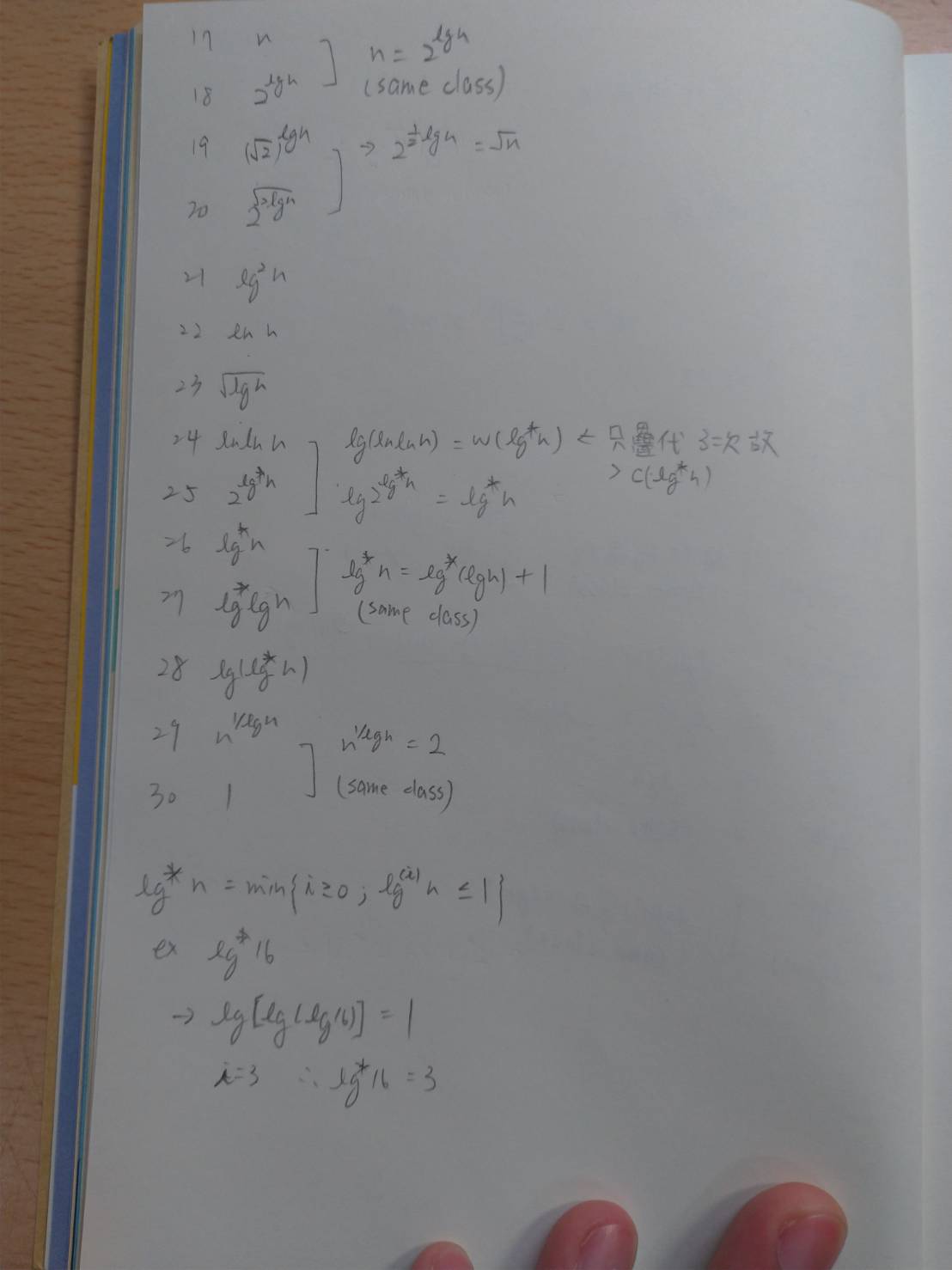
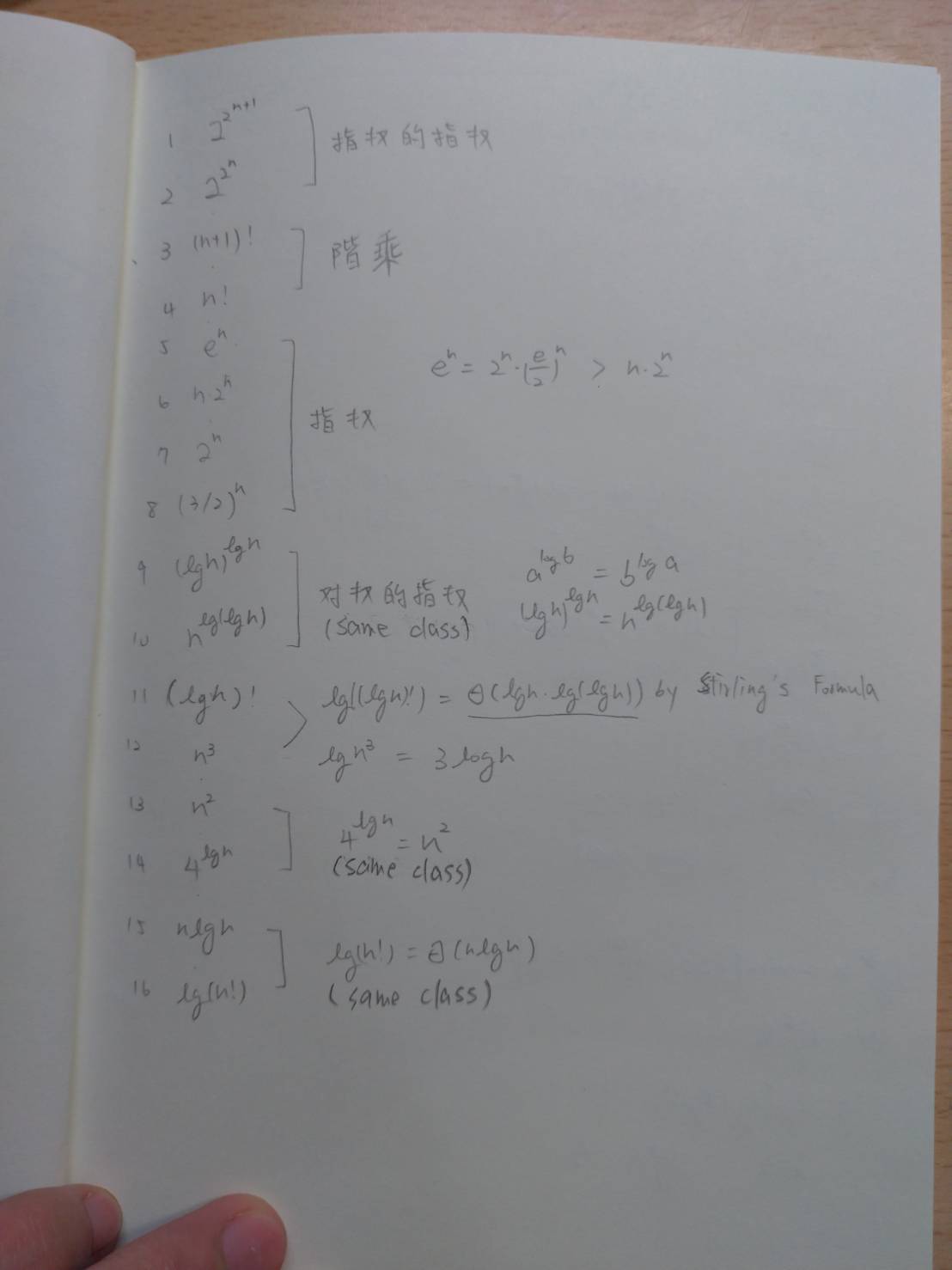
(A)

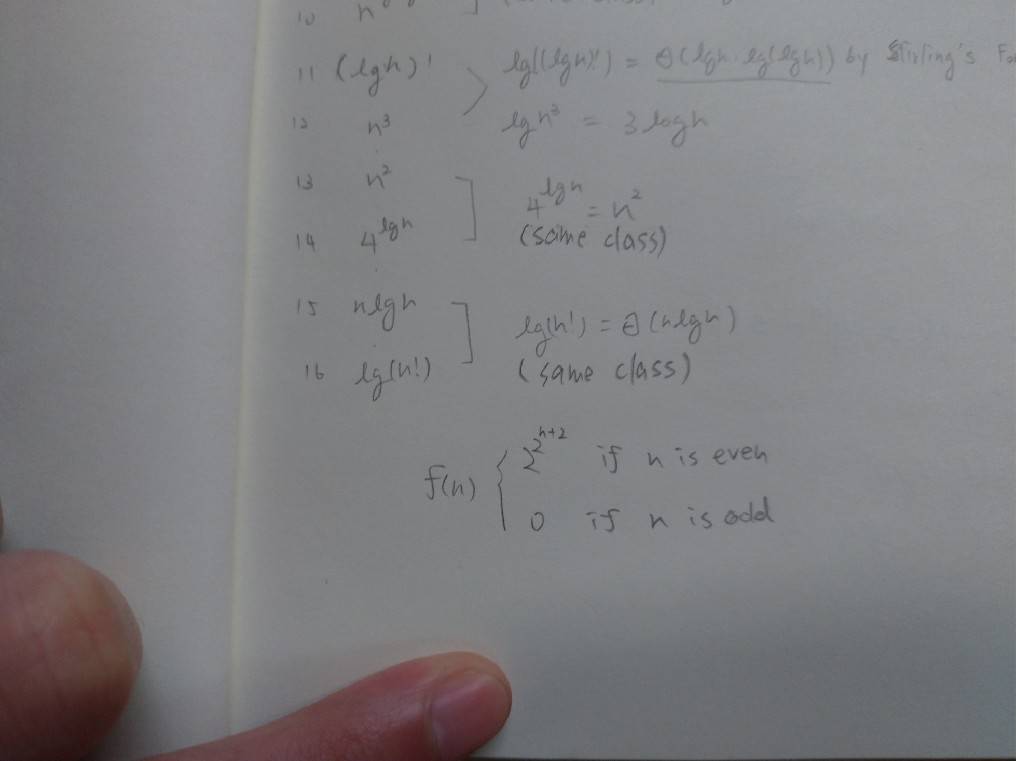
1. ,
2. 和 沒有漸近線，無法比較

,

1. use Stirling’s approximation

Q2. **Problem 3-3 Ordering by asymptotic growth rates**

a. Rank the following functions by order of growth; that is, find an arrangement g1, g2, …, g30 of the functions satisfying g1 = Ω(g2), g2 = Ω(g3), …, g29 = Ω(g30). 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) = Ѳ( g(n) ).  
 About the definition of \*, you can check the textbook.   
  


b. Give an example of a single nonnegative function f(n) such that for all functions gi(n) in part (a), f(n) is neither O(gi(n)) nor Ω(gi(n)  
  
f(n) is oscillation, we can not find a n make ), f(n) O(gi(n)) nor Ω(gi(n).

Q3. **Let f(n) and g(n) be asymptotically positive functions. Prove or disprove each of the following conjectures.**

1. (F) f(n) = O(g(n)) implies g(n) = O(f(n)).

Counter-example:

Let f(n)=1, g(n)=n

> f(n)=O(g(n))=O(n), however g(n)≠O(f(n))=O(1)

b. (F) f(n) + g(n) = Ѳ(min(f(n), g(n))).

Counter-example:

Let f(n)=n, g(n)=n2, thus min(f(n), g(n))=n

However, f(n) + g(n)=n2+n = Ѳ(n2) ≠ Ѳ(n)

c. (T) f(n) = O(g(n)) implies lg(f(n)) = O(lg(g(n))), where lg(g(n)) ≥ 1 and f(n) ≥ 1 for all sufficiently large n.

When f(n) = O(g(n)), which means c\*g(n) ≥ f(n), where c is a constant and f(n) ≥ 1. In the meanwhile, lg(c\*g(n)) ≥ lg(f(n)).

Thus, lg(f(n)) = O(lg(g(n))).

d. (F) f(n) = O(g(n)) implies 2f(n) = O(2g(n)).

Counter-example:

Let f(n)=2n and g(n)=n, so f(n)=O(g(n)) is true.

However, 22n ≠ O(2n) because 22n grows much faster than 2n since limx→∞(2n/22n) = 0.

e. (F) f(n) = O((f(n))2).

Counter-example:

Let f(n)=n1/2, than (f(n))2=n1/4

Thus, f(n) ≠ O((f(n))2) since n1/2 grows faster than n1/4.

f. (T) f(n) = O(g(n)) implies g(n) = Ω(f(n)).

This is true because O-notation is transpose symmetric according to the text book.

g. (F) f(n) = Ѳ(f(n/2)).

Counter-example:

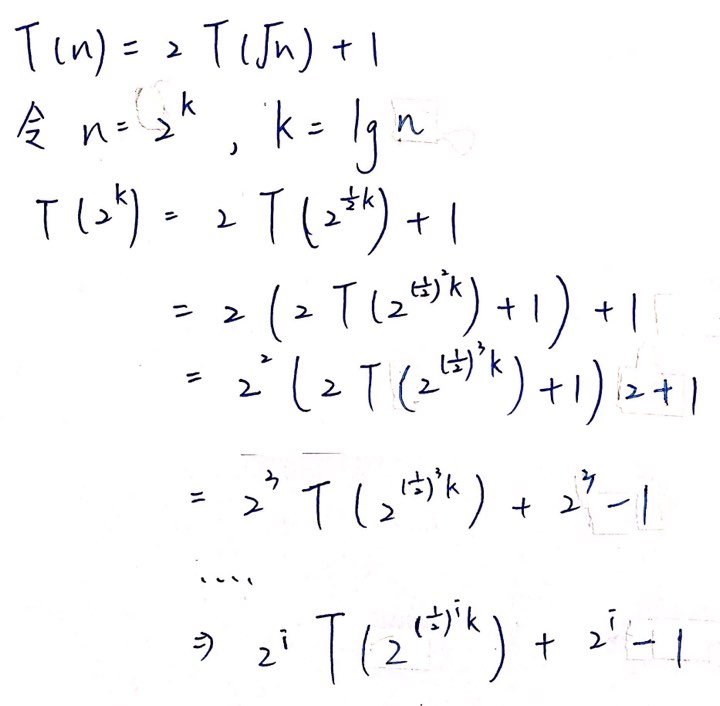
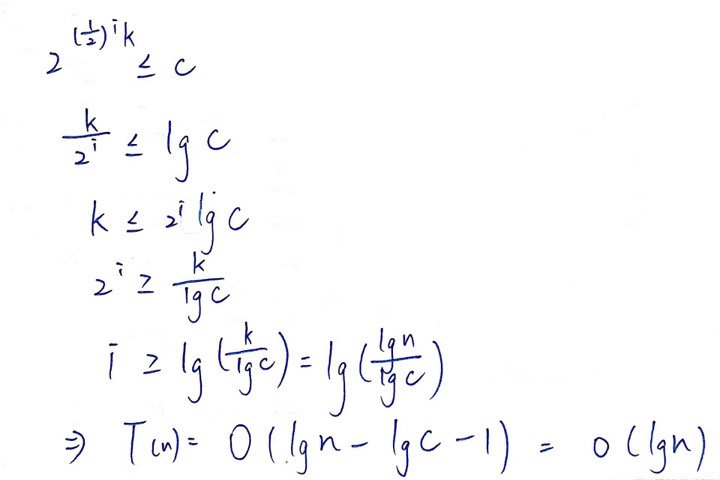
Let f(n)=2n, so f(n/2)=2n/2

However, f(n) ≠ O(2n/2) because 2n grows faster than 2n/2 since limx→∞(2n/2/2n) = 0.

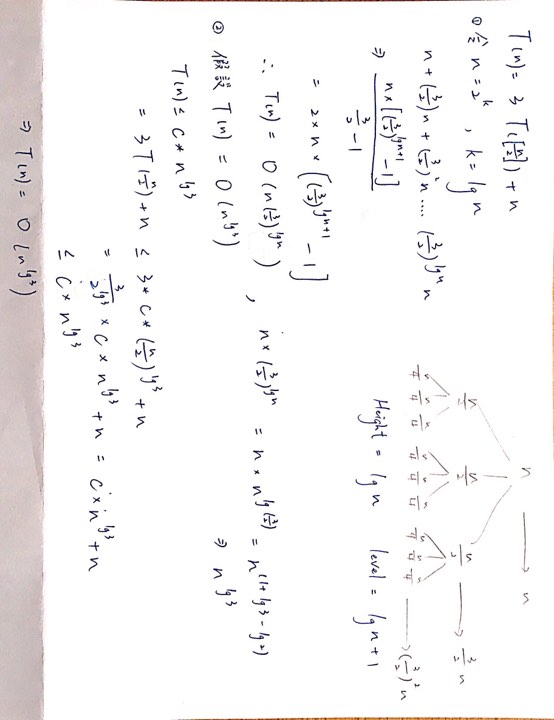
h. (T) f(n) + o(f(n)) = Ѳ(f(n)).

Let g(n) = o(f(n)), than limn→∞(g(n)/f(n)) = 0 according to the definition. So if we try limn→∞[(f(n)+g(n))/f(n)], we can get that limn→∞[1+(g(n)/f(n))] = 1 since g(n)/f(n) part would become 0 when n goes infinite. Thus it implies that f(n) grows as fast as f(n) + g(n). Therefore, f(n) + o(f(n))

Q4. **Solve the recurrence by making a change of variables.** Your solution should be asymptotically tight. Do not worry about whether values are integral.



Q5. **Use a recursion tree to determine a good asymptotic upper bound on the recurrence**  Use the substitution method to verify your answer.



Q6. **Give asymptotic upper and lower bounds for T(n) in each of the following recurrences.**

Assume that T(n) is constant for n≦2. Make your bounds as tight as possible, and justify your answers.

(a) T(n)=2T(n/2)+n³

By Master Theorem , nlogba = n , T(n)=θ(f(n))=θ(n³)

(b) T(n)=T( n)+n

By Master Theorem , nlogba = n0 =1 ,T(n)=θ(f(n))=θ(n)

(c) T(n)=16T(n/4)+n2

By Master Theorem , nlogba = n2 , T(n)= θ(nlogba \*log n)

(d) T(n)=7T(n/3)+n2

By Master Theorem , nlogba = n1…= nk , k<2 , T(n)= θ(f(n))= θ(n2)

(e) T(n)=7T(n/2)+n2

By Master Theorem , nlogba = n2…= nk, k>2,

T(n)= θ(nlogba)= θ(nlog27)

(f) T(n)=2T(n/4)+√n

By Master Theorem , nlogba = √n ,

T(n)= θ(nlogba \*log n)= θ(√n \*log n)

(g) T(n)=T(n-1)+n

Let T(0)=c , c is constant , T(1)=T(0)+1 , T(2)=T(0)+1+2 ……

T(n)=T(0)+1+2+…+n = **c+(n\*(n+1))/2**

* T(n)<=c+n+n+…n=c+n2

-> T(n)=O(n2)

* T(n)>=(n\*(n+1))/2

-> T(n)= Ω(n2)

Hence,T(n)= θ(n2)

(h) T(n)=T(√ n)+1

Let n=22^k  , T(2)=c , c is a constant

**T(n)=22^k-1+1** , let ak =22^k

ak = ak-1 +1= ak-2 +2= a0 +k = **c+log(log n)**

T(n)= θ(log(log n))

Q7. **Give asymptotic upper and lower bounds for T(n) in each of the following recurrences.** Assume that T(n) is constant for sufficiently small n. Make your bounds as tight as possible, and justify your answers.

(a) T(n)=4T(n/3)+n lg n

使用master method，T(n)=aT(n/b)+f(n)，則a=4,b=3,f(n)= n lg n。

logba=log34~=1.2619。

=0，存在ε>0使f(n)=O(nlogba-ε)。T(n)=θ(n^( logba))。

→T(n)=θ(n^( log34))

(b) T(n)=3T(n/3)+n /lg n

T(n)=3T(n/3)+n /lg n

=3[3(T(n/9)+)]+

=3{3[3(T(n/27)+)+]}+

=3kT()+(++........+)，令T(1)=const，n=3k

=n\*T(1) +n( ++........+)

= n\*T(1) +n( ++.......+

= n\*T(1) +n((+.......+))

=n\*T(1) +n(())，帶入k=log3n

= n\*T(1) +n((ln log3n))

=θ(n lg lg n)

(c) T(n)=4T(n/2)+n2

使用master method，T(n)=aT(n/b)+f(n)，則a=4,b=2,f(n)=n2.5。

logba=log24=2。 f(n)=Ω(nlogba+ε)，ε=0.5。T(n)=θ(f(n))。

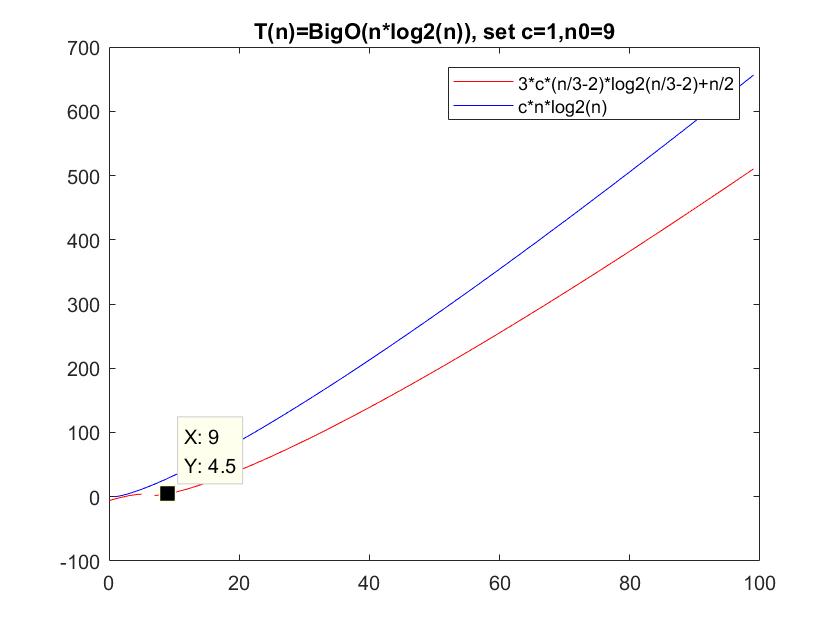
→T(n)=θ(n2.5)。

(d) T(n)=3T(n/3-2)+n/2

使用substitution method。先猜T(n)=O(n lg n)。

欲證明T(n)=O(n lg n)，即存在c>0，n0>=0，當n>n0,T(n)<=c n lg n。

假設 T(n/3-2)<= c (n/3-2) lg (n/3-2)，則 T(n) <=3\*[ c (n/3-2) lg (n/3-2)]+ n/2 <= c n lg n，取c=1,n0=9。T(n)=O(n lg n)得證。



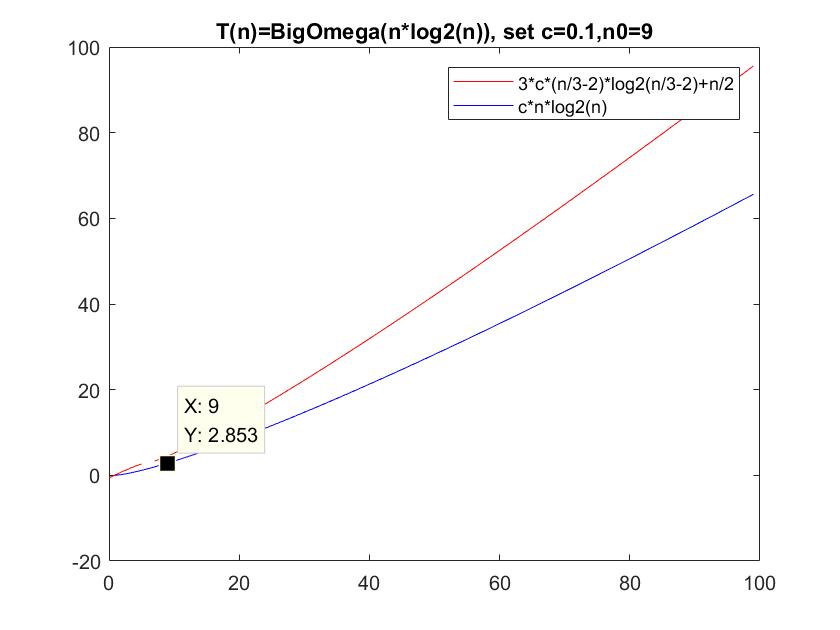
同理，猜T(n)=Ω(n lg n)。

欲證明T(n)= Ω (n lg n)，即存在c>0，n0>=0，當n>n0,T(n)>=c n lg n。

假設 T(n/3-2)>= c (n/3-2) lg (n/3-2)，

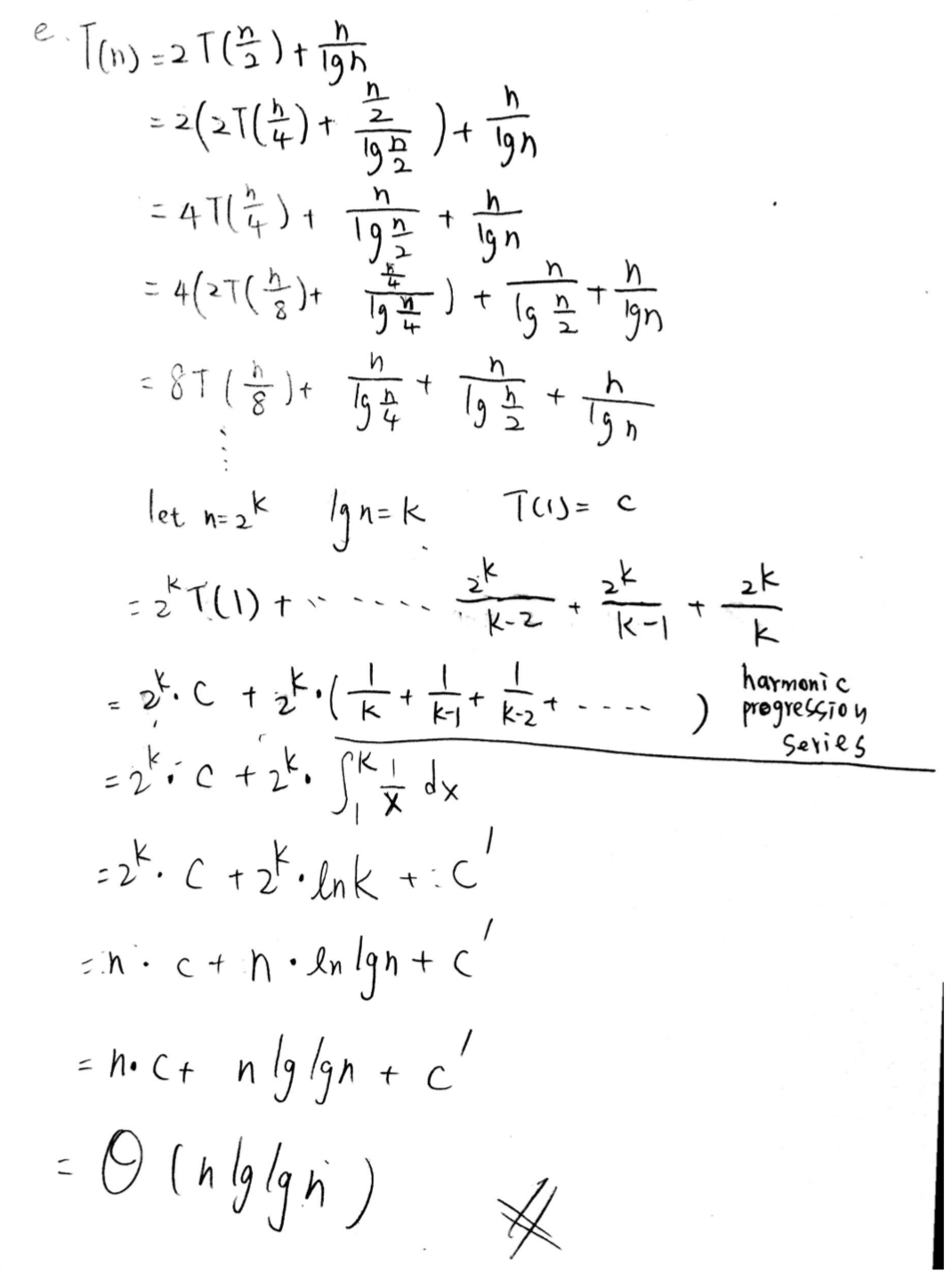
則 T(n) >=3\*[ c (n/3-2) lg (n/3-2)]+ n/2 >= c n lg n，取c=0.1,n0=9。

T(n)= Ω (n lg n)得證。



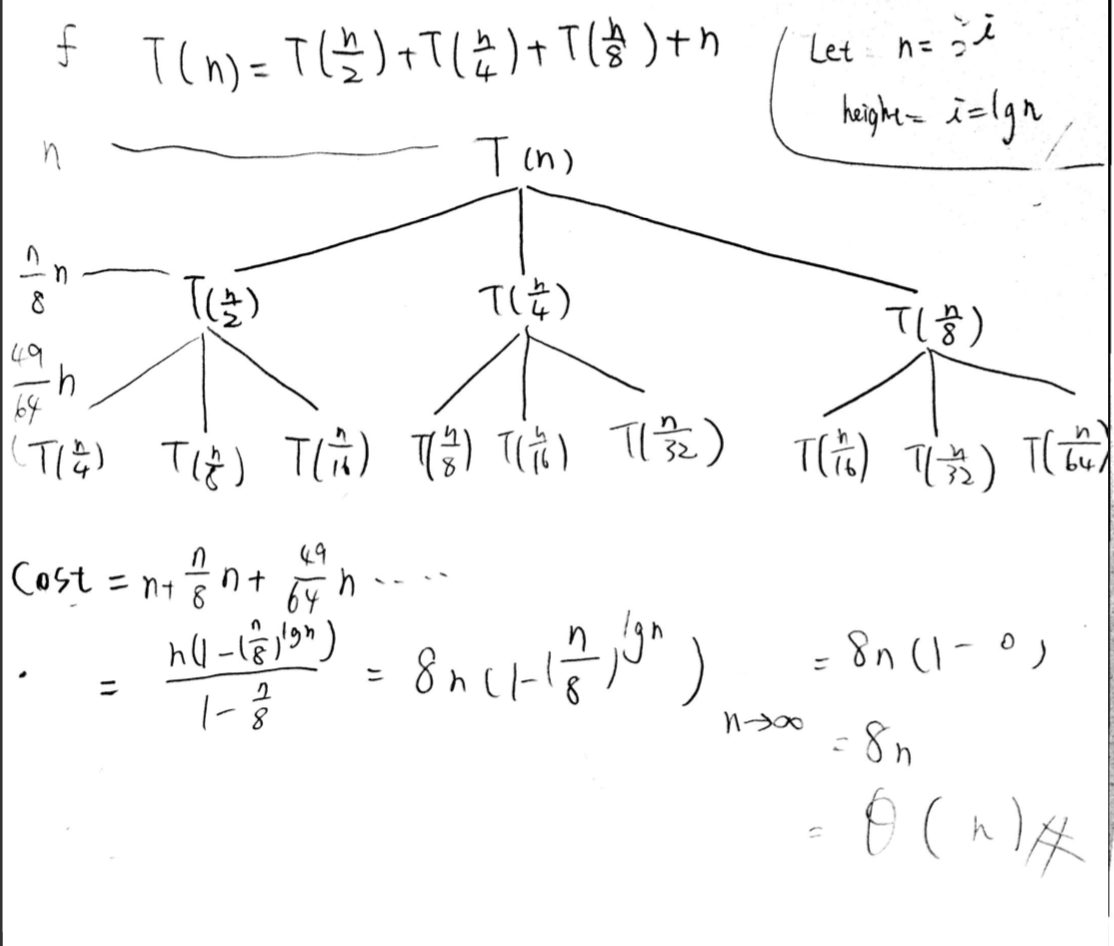
T(n)= Ω (n lg n)且T(n)=O(n lg n)，故得T(n)=θ(n lg n)

(e) T(n)=2T(n/2)+n/ lg n



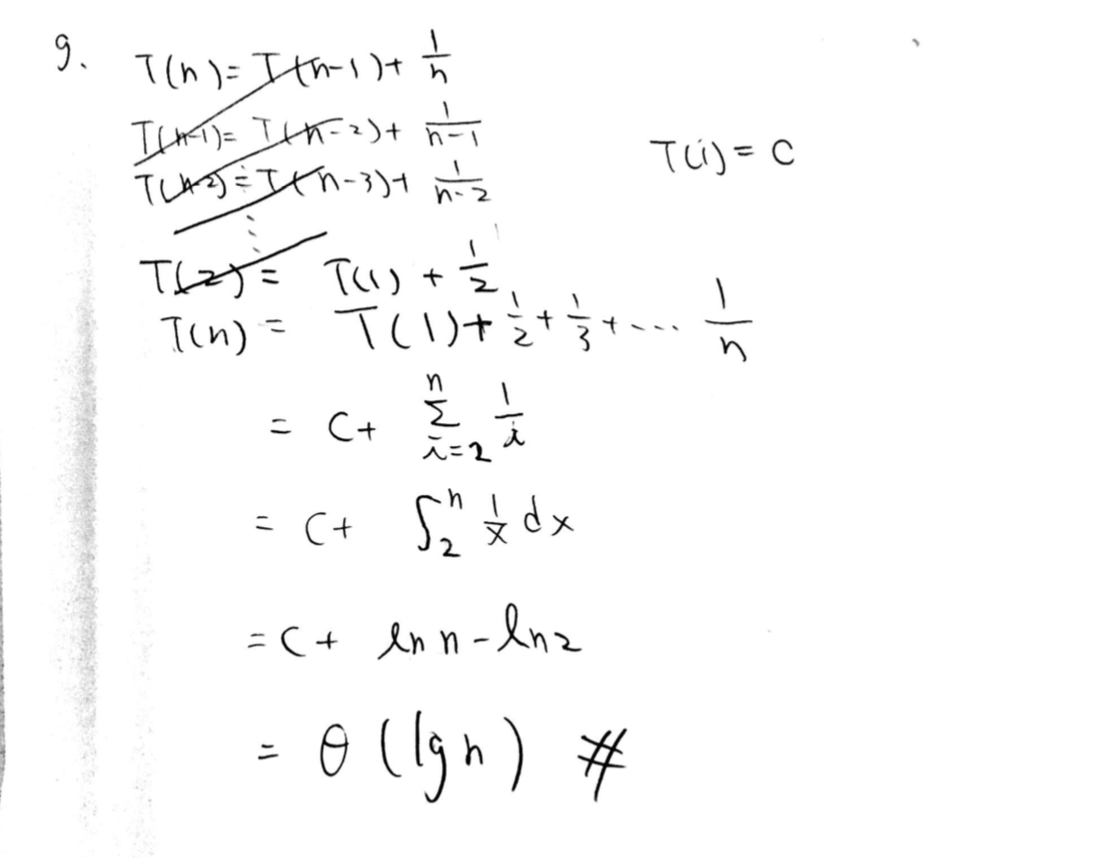
T(n)=θ(n lg(lg n))

(f) T(n)=T(n/2)+T(n/4)+T(n/8)+n



T(n)=θ( n)

(g) T(n)=T(n-1)+n



T(n)=θ( lg n)

(h) T(n)=T(n-1)+lg n

1. T(n)=T(n-2)+1/lg n

(j) T(n)=√nT(n-1)+n

其中紅色部分會從)，所以可以得出

其中By Stirling’sapproximation