$100^2 - 99^2 = (100+99)(100-99)$

$(a+b)$.

Wk 1 - Tutorial

1) Sort the following functions in increasing order of asymptotic growth

~~n~~, n^3, ~~$n \log n$~~, n^n, $\frac{3^n}{n^2}$, ~~$n!$~~ ~~$\sqrt{n}$~~, 2^n

- ~~$O(n)$~~
- ~~$O(n \log n)$~~ $O(\sqrt{n})$ 1
- $O(n)$ 2
- $O(n \log n)$ 3
- $O(n!)$ 4
- $O(n^3)$ 5
- $O(2^n)$ 6
- $O(\frac{3^n}{n^2})$ 7
- $O(n^n)$ 8

2) $\log\log n = x$ $10^x = \log n$ $(10)^{10^x} = n$ $(10^{10})^x = n$ $\log_{10^{10}} n$

$\log(n!)$

$2^{\log\log n}$ 2^x

$n^{\frac{1}{\log n}} = (10^A)^{\frac{1}{A}} = 10^1 = 10$

$1 \xi_0 O(n^{\frac{1}{\log n}}) = O(1)$

$2 \xi_0 O(\log\log n) = \log A$

$3 \xi_0 (2^{\log\log n}) = 2^{\log A}$

$4 \xi_0 O((\log n!) = A + \log(A-1)!)$

$\log n = A$

$10^A = n$

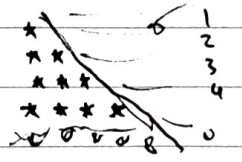
$\log(n!) = \log(n \times (n-1)!) = \log n + \log(n-1)!$ $1000000 = 10^6$

log $2^{\log A} = x$ $\frac{1}{\log_{A10}}$ 2^x $2^{\log\log n}$ $O \log(n!)$

$2^{\log_{1010} n}$ $O \log(n!)$

3)

```
function STARS(n):
  for i ← 1; i ≤ n; i++ do
    Print '*' i times
```



$O(n^2)$

$\Omega(\frac{n^2}{2})$ or ~~$\Omega(\sum_{i=0} i)$~~

for any one step

$t(n) = \sum_{i=0}^{n} i$

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

$= \frac{n^2}{2} + \frac{n}{2}$

0 0

1 1

2 4

5 a)

```
function summing-up(A):
    C ← new matrix (A×A)
    for i ← 0; i<n; i++ do
        for j ← i; j<n; j++ do
            average of entries A[i:j]
            C[i,j] ← ⊖
```

$\frac{A[i]+A[i+1]\dots A[j]}{j-i+1}$ where $0 \le i \le j < n$

$O(n^3)$

$\Omega(n^3)$

$i < \frac{1}{4}n$

$j > \frac{3}{4}n$

$0 \sim \frac{1}{4}n$

$\frac{3}{4}n \sim n$

$= \frac{1}{16}n^2 \times n = \Omega(n^3)$

6

$i=1 \quad j=3$

$n=3 \quad 0 \le i \le j < 3$

i \ j	1	2	3
1	1		
2	1.5	2	
3	2	2.5	3

0 0 11 2

0 1 12

0 2

$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

[1, 2, 3] n=3,

[1, 2, 3, 4, 5]

15

1
3 2
6 5
10 9

A = [1, 2, 3, 4, 5] ~~B = []~~

n = A's length

~~for i=0; i<n; i++:~~

~~for~~ sum = 0

for i ~ n:

sum += i

B[i] = sum

0 1 3 6

B = [1, 3, 6, 10, 15]

$\frac{6-1}{2} = 2.5$

i=1

i=2

for i = 0 ~ n:

for j = i ~ n: B[j] - B[i-1]

C[i][j] = $\frac{B[j]-B[i-1]}{j-i+1}$

7) if $f(n) = O(g(n))$ ~~$\therefore f(n) = k$~~

$g(n) = O(h(n))$

Assume f, g, n is polynomial

(answer on back) ~~ass~~

$f(n)$'s leading term is equal to $g(n)$ with coefficient k.

$g(n)$'s leading term is $g(n)$ itself.

$g(n)$'s leading term is $O(h(n))$.

$\therefore$ ~~$g(n)$~~ $= O(h(n))$

$\therefore f(n) =$ ~~k~~ $k O(h(n))$

$= O(h(n))$

8)

```
j ← 0
for i = 0 ; i < n ; i++ do
    while j ≥ 1 and condition:
        j ← j - 1
    j = j + 1
return j
```

n = 3

i=0 1 step (j += 1)

i=1 j -= 1 | (j = 0)

$O(n^2)$

for loop: n steps

depends on ~~condition~~

~~once i~~ from i = 1 to n, the while loop may repeat (k) times because 0 ≤ k ≤ 1

if condition:

    j stays same

else:

    j increments (without taking step).

the next loop would iterate only

7)

Assume $f(n) = O(g(n))$

Assume $g(n) = O(h(n))$

By definition there exists no constant c where $cf(n) > g(n)$

$\therefore cf(n) \not< g(n)$

Also by the same definition

$\therefore d\,g(n) \not< h(n)$ (d is a constant)

$cd\,f(n) \not< dg(n) \not< h(n)$ cd is constant c×d

$cd\,f(n) < h(n)$

$(\,f(n) < h(n)$ (= constant (replaces c×d)

applies for any c

$\therefore f(n) = O(h(n))$

9)

```
def (k, str) :
    consecutive = 0 ; last = k ~ str[0]
for i = 1 ; i < str.length ; i++ do
    if str[i] == str[i-1] do
```

```
if k = 0 :
    return True
else
```

(refer to github tut_q9.py)

number of consecutive characters required for this function to return True.

```
function consecutive_k ( k, string ) :
    if k < 1 or string's length < 1
        return " invalid arguments"
    consecutive_num = 1
    for i from 1 to len(string) do
        if string[i] == string[i-1]
            consecutive_num += 1
        else
            consecutive_num = 1   # resetting
        if ( consecutive_num >= k]
            break
    if con_k >= K
        r True
    else
        return False
```

10)

~~Conten~~

~~numbers~~: find_duplicate(A)

sort A

iterate A if ~~elem~~ A[i] = = A[i+1] applies

for any i from 0 to len(A) - 2.

or ~~brute~~ force

or hashing.