

What does $O(< expr >)$ mean?

1

What does $\Theta(< expr >)$ mean?

2

What does $\Omega(< expr >)$ mean?

3

*What are the best, average and worst case complexities of **Bubble Sort**?*

4

*What are the best, average and worst case complexities of **Merge Sort**?*

5

Give pseudo code for merging 2 sorted lists, as part of merge sort.

6

Give pseudo code for MergeSort(L).

7

*What are the best, average and worst case complexities of **Quick Sort**?*

8

The complexity (i.e. space/running time) has the complexity proportional to $\langle \text{expr} \rangle$.

The complexity (i.e. running time/space) is bounded by the $\langle \text{expr} \rangle$.

2

1

Best: $O(n)$,
Average: $O(n^2)$,
Worst: $O(n^2)$

The complexity (i.e. running time/space) is at least by the $\langle \text{expr} \rangle$.

4

3

```
Merge( $L_1$ ,  $L_2$ )
  if  $L_1 = []$  return  $L_2$ 
  if  $L_2 = []$  return  $L_1$ 
   $x_1 = L_1[0]$ 
   $x_2 = L_2[0]$ 
   $L'_1 = L_1[1 : |L_1| - 1]$ 
   $L'_2 = L_2[1 : |L_2| - 1]$ 
  if  $x_1 \leq x_2$ 
    return  $[x_1] + \text{Merge}(L'_1, L_2)$ 
  return  $[x_2] + \text{Merge}(L_1, L'_2)$ 
```

Merge two sorted lists

Best: $O(n \log_2 n)$,
Average: $O(n \log_2 n)$,
Worst: $O(n \log_2 n)$

6

5

Best: $O(n \log_2 n)$,
Average: $O(n \log_2 n)$,
Worst: $O(n^2)$

```
MergeSort( $L$ )
  if  $|L| \leq 1$ 
    return  $L$ 
  Split  $L$  into roughly equal halves,  $L_l$  and  $L_r$ 
  return Merge(MergeSort( $L_l$ ), MergeSort( $L_r$ ))
```

MergeSort(L)

8

7

<p><i>What would the pseudo code be for Quick Sort?</i></p> <p>9</p>	<p><i>Say that the input represents a positive integer, x, what is the size of n?</i></p> <p>10</p>
<p><i>What does it mean by $O(1)$?</i></p> <p>11</p>	<p><i>What is the minimum time for any sorting algorithm that uses only number comparisons?</i></p> <p>12</p>
<p><i>What would the pseudo code be for Euclid's algorithm?</i></p> <p>13</p>	<p><i>What would the pseudo code be for Fast Modular Exponentiation?</i></p> <p>14</p>
<p><i>What are some of the advantages of ElGamal encryption?</i></p> <p>15</p>	<p><i>What is the basic procedure for an encryption and decryption using publik key cryptography if Alice wants to send a message to Bob?</i></p> <p>16</p>

$\lfloor \log_b x \rfloor + 1$ Where b is the number representation,
usually binary (so 2).

```
quicksort(L)
  if length of L ≤ 1
    return L
  remove the first element, x, from L
  L≤ := elements of L less than or equal to x
  L> := elements of L greater than x
  Ll := quicksort(L≤)
  Lr := quicksort(L>)
  return Ll + [x] + Lr
```

Quick Sort

10

9

$n \log_2 n$

It takes a constant time, no matter the amount of
data, to perform the operation.

12

11

```
fme(a,b,k)
  d = a
  e = b
  s = 1
  While e > 0
    if e is odd
      s = (s.d) mod k
      d = d2 mod k
      e = ⌊e/2⌋
  return s
```

Fast Modular Exponentiation

14

```
// Assume a ≥ b
hcf(a,b)
  if b = 0
    return a
  r = a mod b
  return hcf(b,r)
```

Euclid's algorithm

13

Alice generates a private random integer a and Bob
generates a private random integer b
Alice generates her public value $g^a \bmod p$
Bob generates his public value $g^b \bmod p$
Alice computes $g^{ab} = (g^a)^b \bmod p$
Bob computes $g^{ba} = (g^b)^a \bmod p$
Now they have a shared secret k since $k = g^{ab} = g^{ba}$

Sender Verification
Private key remains with owner
Public key is freely distributable
No secret channel needed at any point
No need for pre-shared keys

16

15

<p><i>Describe public key generation in ElGamal encryption using p as the Prime Modulus and g as the Primitive root (as described in the COMP26120 lab)</i></p> <p>17</p>	<p><i>Consider the equation $a^x = y \bmod p$. If a is a primitive root modulo p, then for every $y(1 \leq y < p)$, such an $x(1 \leq x < p)$ exists. What is x?</i></p> <p>18</p>
<p><i>The is the inverse of exponentiation.</i></p> <p>19</p>	<p><i>Why can the private key not, in practice, be recovered from the public key when p is large?</i></p> <p>20</p>
<p><i>What is one way you can argue correctness of Euclid's algorithm?</i></p> <p>21</p>	<p><i>What would half the correctness proof be for Euclid's algorithm?</i></p> <p>22</p>
<p><i>$(a.b) \bmod k =$ </i></p> <p>23</p>	<p><i>Let p be a prime number. What is meant by a primitive root modulo p?</i></p> <p>24</p>

X is the **discrete logarithm** of y with base a ,
modulo p .

18

Generate an x randomly from $1 \leq x < p$. This is the
private key to be kept secret. Now compute h , part
of the public key, using
$$h = g^x \pmod{p}$$

(This is modular exponentiation)
Note that the full public key is a combination of $(h,$
 $g, p)$

17

To calculate a public key, y , a private key, x is
needed. The equation for modular exponentiation
can be used: $y = g^x \pmod{p}$

It is considered a one-way function - easy to
compute, hard to invert. For a large p , the only way
to figure out the private key would be to use brute
force, which would take a large amount of time.

20

The discrete logarithm is the inverse of
exponentiation.

19

As $r = a \pmod{b}$, $\exists q$ such that $a = bq + r$, $\therefore r = a - bq$.
Suppose x is a factor of a and b , then $\exists y$ and z such
that $a = xy$, $b = xz$.

Hence: $r = xy - xzq$, $r = x(y - zq)$.
 $\therefore x$ is a factor of r (and also of b and r).

22

Let $r = a \pmod{b}$. $\text{hcf}(a, b) = \text{hcf}(b, r)$ because all
factors of a and b are also factors of b and r and
vice versa. If they have the same factors, they have
the same highest common factor.

21

The numbers r_x between 1 and $p - 1$ that, when
raised by the numbers between 1 and $p - 1$ compute
all the numbers between 1 and $p - 1$ in some order
with no repetitions.

24

$$(a.b) \pmod{k} = (a \pmod{k}.b \pmod{k}) \pmod{k}$$

23

<p><i>What does saying that algorithm A runs in time g mean?</i></p> <p>25</p>	<p><i>What is a permutation of a set?</i></p> <p>26</p>
<p><i>What do we mean by a composition of two permutations?</i></p> <p>27</p>	<p><i>What is the number of possible permutations on an n-element set?</i></p> <p>28</p>
<p><i>In the context of a permutation, what do we mean by a transposition?</i></p> <p>29</p>	<p><i>Convert this pair of simultaneous equations into matrix form</i></p> $\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 &= b_2 \end{aligned}$ <p>30</p>
<p><i>What is the determinant of the matrix:</i></p> $\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ <p>31</p>	<p><i>What is an upper triangular matrix and how do you calculate its determinant?</i></p> <p>32</p>

A 1-to-1 map of the set onto itself. In basic terms,
it is a set mapped to another order of itself. i.e
 $[0, 1, 2, 3, 4] \mapsto [2, 4, 1, 0, 3]$

Given an input of size n , the number of operations
executed by A is bounded above by $g(n)$.

26

25

$n!$

The composition is the product of two permutations,
 α and β , on a set n , given by $\alpha \cdot \beta(n)$ or $\beta(\alpha(n))$

28

27

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$\sigma(k) = \begin{cases} j & \text{if } k = i \\ i & \text{if } k = j \\ k & \text{ow.} \end{cases}$$

30

29

It is a matrix where all of its entries below the
diagonal are zero.

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n,n} \end{pmatrix}$$

Its determinant is calculated by taking the product of
the entries on the diagonal. i.e $a_{1,1} \cdot a_{2,2} \cdot \dots \cdot a_{n,n}$

$a_1a_4 - a_2a_3$
Often denoted as:

$$\begin{vmatrix} a_1 & a_2 \\ a_3 & a_4 \end{vmatrix}$$

The original system of equations to which the matrix
corresponds only has a unique solution if the
determinant is non-zero.

32

31

Which 4 operations have no effect on a matrix's determinant?

Transposing two rows
Transposing two columns
Adding a multiple of one row to another
Adding a multiple of one column to another
Also note that if all entries in any row or column
are 0 then the determinant is 0