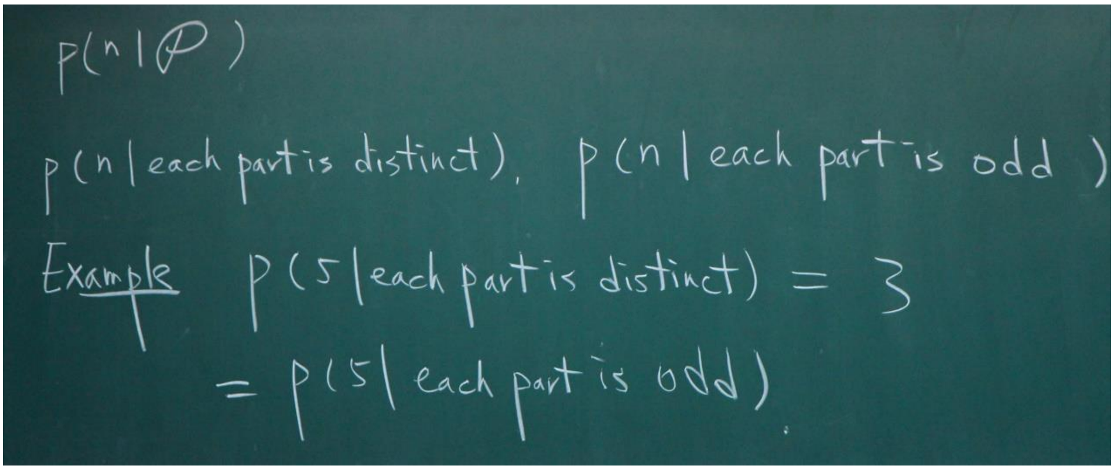
【10920 趙啟超教授離散數學 / 第 17 堂版書】



$p(n \mid \text{each part is distinct})$, $p(n \mid \text{each part is odd})$

Example $p(5 \mid \text{each part is distinct}) = 3$
$= p(5 \mid \text{each part is odd})$.

Example

$$\begin{aligned} 8 &= 8 \\ &= 7+1 \\ &= 6+2 \\ &= 5+3 \\ &= 5+2+1 \\ &= 4+3+1 \end{aligned}$$

$\therefore p(8 \mid \text{each part is distinct}) = 6$

$$\begin{aligned} 8 &= 7+1 \\ &= 5+3 \\ &= 5+1+1+1 \\ &= 3+3+1+1 \\ &= 3+1+1+1+1+1 \\ &= 1+1+1+1+1+1+1+1 \end{aligned}$$

$\therefore p(8 \mid \text{each part is odd}) = 6$

$p(n \mid \mathcal{P})$

$p(n \mid \text{each part is distinct})$, $p(n \mid \text{each part is odd})$

Example $p(5 \mid \text{each part is distinct}) = 3$
$= p(5 \mid \text{each part is odd})$.

Example

$$\begin{aligned} 8 &= 8 \\ &= 7+1 \\ &= 6+2 \\ &= 5+3 \\ &= 5+2+1 \\ &= 4+3+1 \end{aligned}$$

$\therefore p(8 \mid \text{each part is distinct}) = 6$

$$\begin{aligned} 8 &= 7+1 \\ &= 5+3 \\ &= 5+1+1+1 \\ &= 3+3+1+1 \\ &= 3+1+1+1+1+1 \\ &= 1+1+1+1+1+1+1+1 \end{aligned}$$

$\therefore p(8 \mid \text{each part is odd}) = 6$

Is this a coincidence? No.

<u>Property</u> $p(n \mid \text{each part is distinct})$
$= p(n \mid \text{each part is odd})$.

<u>Proof</u> Let the generating functions for $p(n \mid \text{each part is distinct})$ and $p(n \mid \text{each part is odd})$ be $P_d(x)$ and $P_o(x)$, respectively.

Is this a coincidence? No.

<u>Property</u> $p(n \mid \text{each part is distinct})$
$= p(n \mid \text{each part is odd})$.

<u>Proof</u> Let the generating functions for $p(n \mid \text{each part is distinct})$ and $p(n \mid \text{each part is odd})$ be $P_d(x)$ and $P_o(x)$, respectively.

Then

$$
\begin{aligned}
P_d(x) &= (1+x)(1+x^2)(1+x^3)\cdots\cdots \\
&= \frac{1\cancel{-x^2}}{1-x}\cdot\frac{1\cancel{-x^4}}{1\cancel{-x^2}}\cdot\frac{1\cancel{-x^6}}{1-x^3}\cdot\cdots\cdots \\
&= \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots\cdots} \\
&= (1+x+x^2+\cdots)(1+x^3+x^6+\cdots)(1+x^5+x^{10}+\cdots)\cdots \\
&= P_o(x).
\end{aligned}
$$

Is this a coincidence? No.

<u>Property</u> $p(n \mid \text{each part is distinct})$
$= p(n \mid \text{each part is odd})$

<u>Proof</u> Let the generating functions for $p(n \mid \text{each part is distinct})$ and $p(n \mid \text{each part is odd})$ be $P_d(x)$ and $P_o(x)$, respectively.

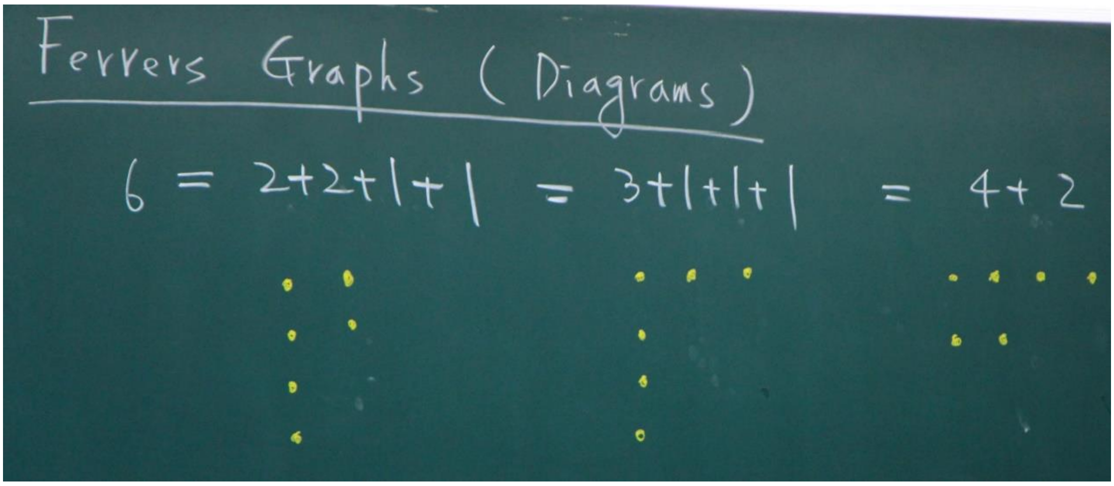
Then
$$
\begin{aligned}
P_d(x) &= (1+x)(1+x^2)(1+x^3)\cdots\cdots \\
&= \frac{1-\cancel{x^2}}{1-x}\cdot\frac{1-\cancel{x^4}}{\cancel{1-x^2}}\cdot\frac{1-\cancel{x^6}}{1-x^3}\cdots\cdots \\
&= \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots} \\
&= (1+x+x^2+\cdots)(1+x^3+x^6+\cdots)(1+x^5+x^{10}+\cdots)\cdots \\
&= P_o(x).
\end{aligned}
$$

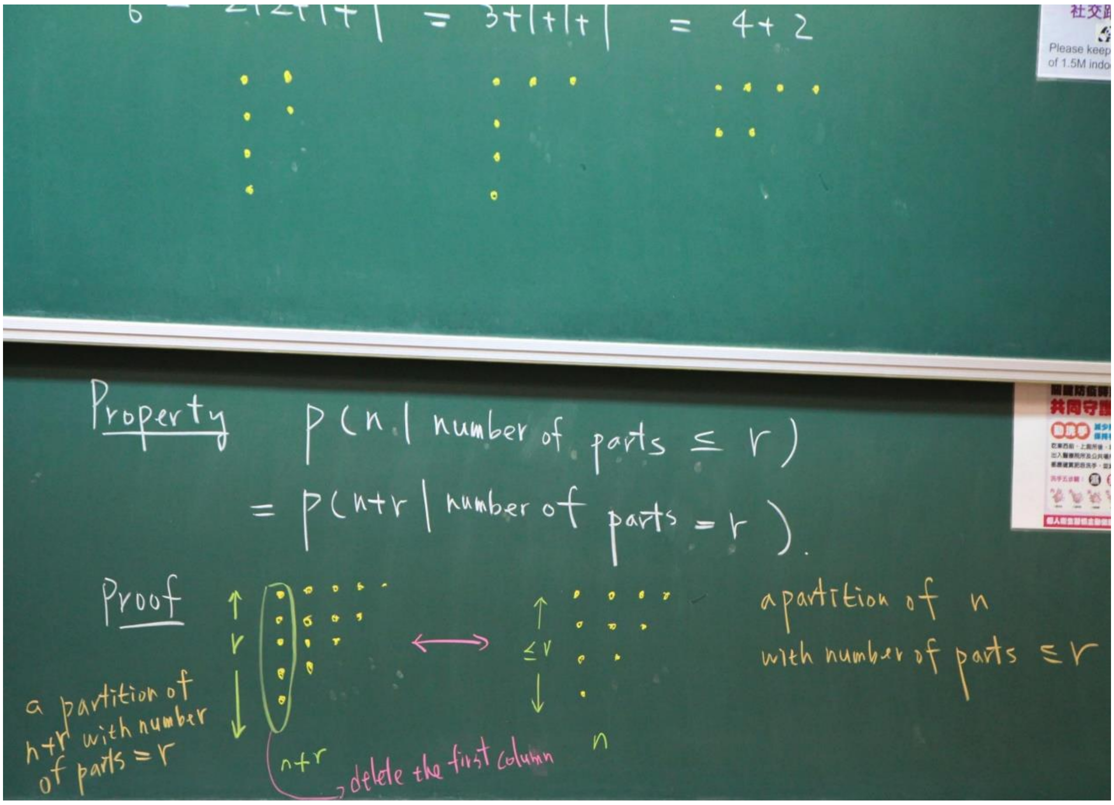
Is this a coincidence? No.

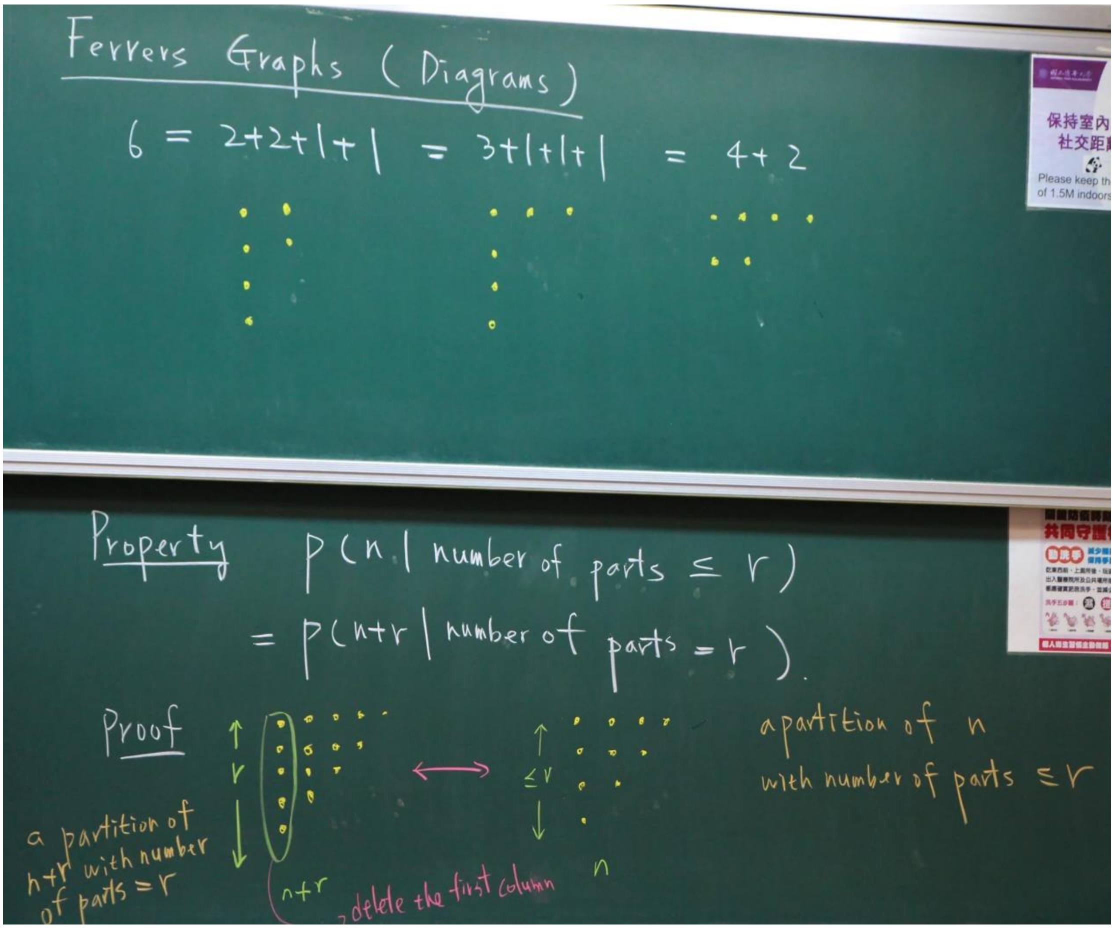
<u>Property</u> $p(n \mid \text{each part is distinct})$
$= p(n \mid \text{each part is odd})$

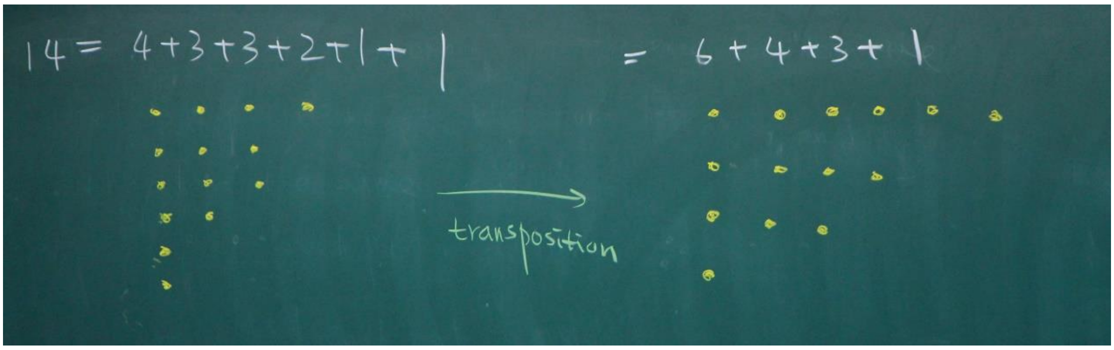
<u>Proof</u> Let the generating functions for $p(n \mid \text{each part is distinct})$ and $p(n \mid \text{each part is odd})$ be $P_d(x)$ and $P_o(x)$, respectively.

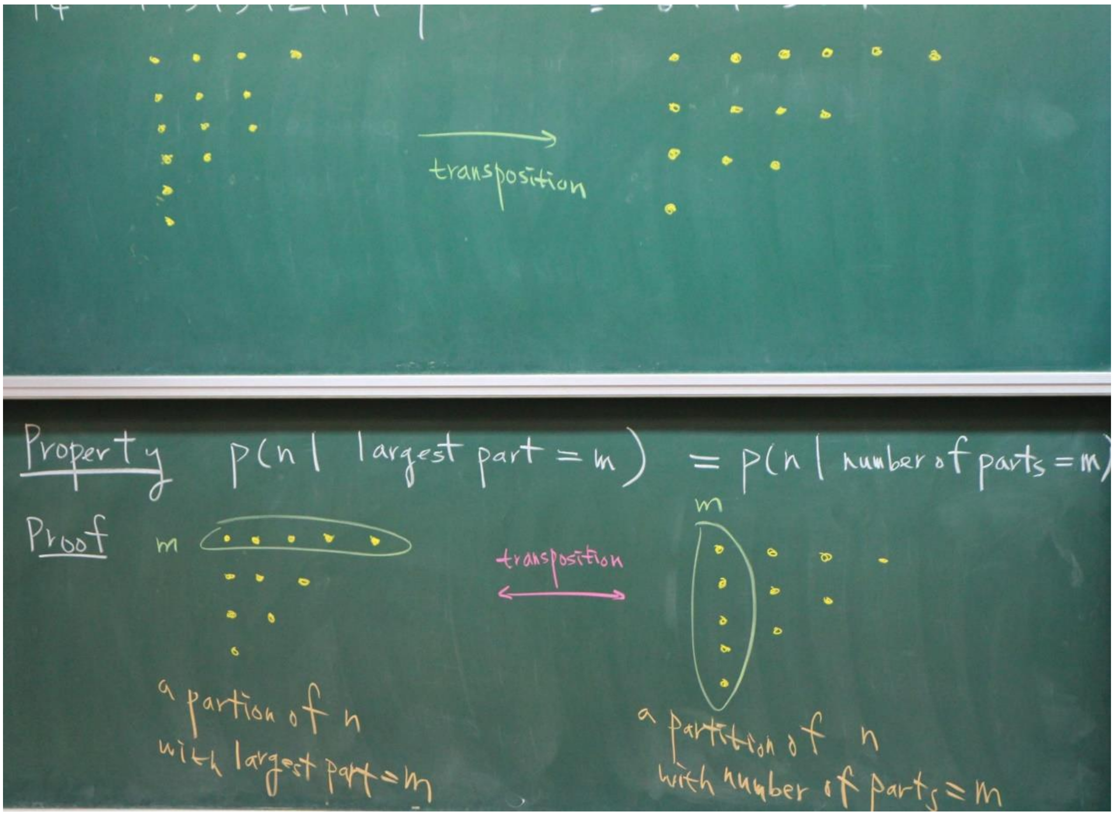
Then
$$
\begin{aligned}
P_d(x) &= (1+x)(1+x^2)(1+x^3)\cdots\cdots \\
&= \frac{1-\cancel{x^2}}{1-x}\cdot\frac{1-\cancel{x^4}}{\cancel{1-x^2}}\cdot\frac{1-\cancel{x^6}}{1-x^3}\cdots\cdots \\
&= \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots} \\
&= (1+x+x^2+\cdots)(1+x^3+x^6+\cdots)(1+x^5+x^{10}+\cdots)\cdots \\
&= P_o(x).
\end{aligned}
$$

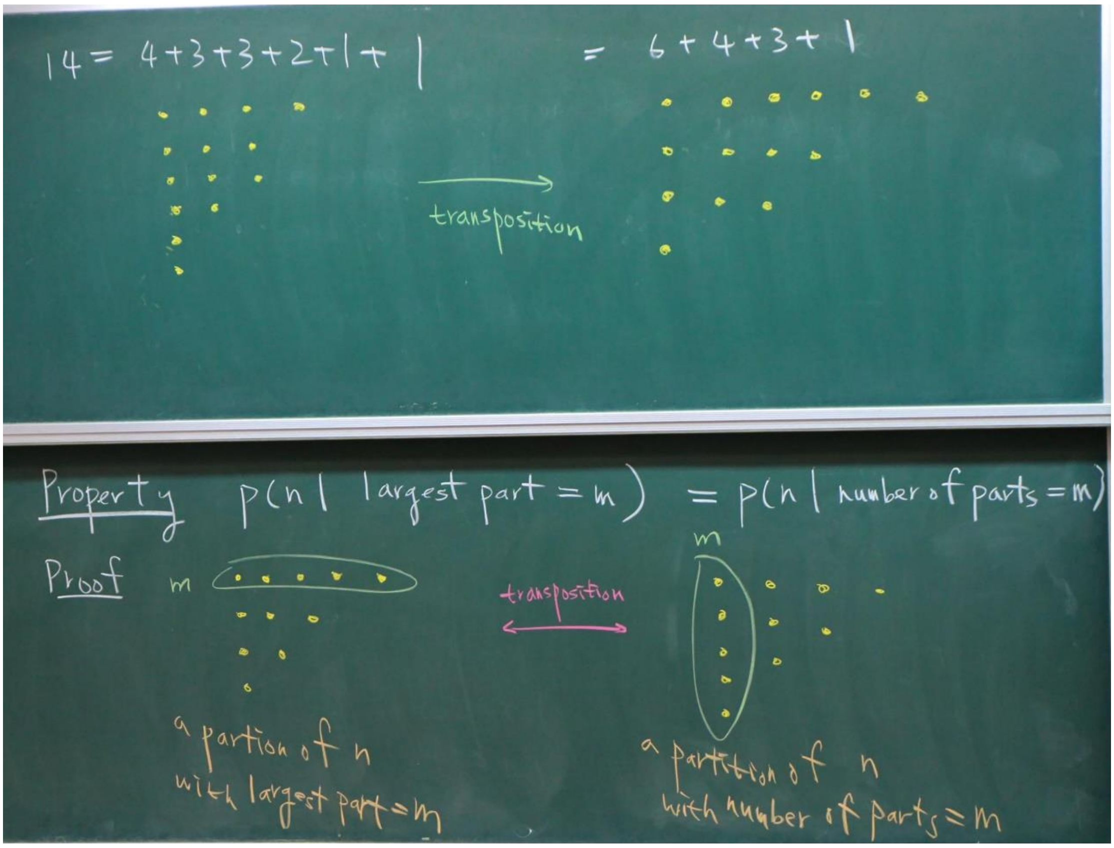

Ferrers Graphs (Diagrams)
6 = 2+2+1+1 = 3+1+1+1 = 4+2


= 3+1+1+1 = 4+2
Property p(n | number of parts ≤ r)
= p(n+r | number of parts = r).
Proof
a partition of n+r with number of parts = r
r
n+r
delete the first column
≤r
n
a partition of n with number of parts ≤ r


Ferrers Graphs (Diagrams)
6 = 2+2+1+1 = 3+1+1+1 = 4+2
Property p(n | number of parts ≤ r)
= p(n+r | number of parts = r).
Proof
r
a partition of n+r with number of parts = r
n+r
delete the first column
≤r
n
a partition of n with number of parts ≤ r


14 = 4+3+3+2+1+1 = 6+4+3+1
transposition


transposition
Property
p(n | largest part = m) = p(n | number of parts = m)
Proof
m
transposition
m
a partion of n
with largest part = m
a partition of n
with number of parts = m


14 = 4+3+3+2+1+1 = 6+4+3+1
transposition
Property
p(n | largest part = m) = p(n | number of parts = m)
Proof
m
transposition
m
a partion of n
with largest part = m
a partition of n
with number of parts = m

$$= \frac{1}{(1-x)(1-x^3)(1-x^5)\cdots}$$

$$(1+x+x^2+\cdots)(1+x^3+x^6+\cdots)(1+$$

$$= P_o(x).$$

There is a one-to-one correspondence between the sets of partitions of the two kinds, and so they have the same cardinality. ∎

Complexity of Algorithms

An algorithm is a list of precise instructions designed to solve a particular type of problem — not just one special case.

For analysis of an algorithm:

1. Describe the algorithm precisely.
2. Define the size of an instance n
3. Calculate $f(n)$, the number of operations required.

Complexity of Algorithms

An algorithm is a list of precise instructions designed to solve a particular type of problem — not just one special case.

For analysis of an algorithm:

1. Describe the algorithm precisely.
2. Define the size of an instance n.
3. Calculate $f(n)$, the number of operations required.

The complexity of a given algorithm is the number $f(n)$ of operations (of a specified kind) required by the algorithm for an instance of size n.

required by the algorithm for an instance of size n.

Example Assume one million operations per second

MIPs (million instructions per second)

$f(n)$	$n=20$	$n=40$	$n=60$
n	0,00002 sec	0,00004 sec	0,00006 sec
n^2	0,0004 sec	0,0016 sec	0,0036 sec
n^3	0,008 sec	0,064 sec	0,216 sec
2^n	1,0 sec	12,7 days	366 centuries

The complexity of a given algorithm is the number $f(n)$ of operations (of a specified kind) required by the algorithm for an instance of size n.

Example Assume one million operations per second

MIPs (million instructions per second)

$f(n)$	$n=20$	$n=40$	$n=60$
n	0,00002 sec	0,00004 sec	0,00006 sec
n^2	0,0004 sec	0,0016 sec	0,0036 sec
n^3	0,008 sec	0,064 sec	0,216 sec
2^n	1,0 sec	12,7 days	366 centuries

Def Let f be a function from $\mathbb{N}$ to $\mathbb{N}$.
We say that $f(n)$ is $O(g(n))$
if there exist positive integers k and n_0
such that

$$f(n) \le k\, g(n)$$

for all $n \ge n_0$.

$$f(n) \le k\, g(n)$$

for all $n \ge n_0$.

Remark The symbol $O(g(n))$ is pronounced "big-oh of $g(n)$."

Example $f(n) = 3n^3 - 20n^2 + 5n - 11$

$$\le (|3| + |20| + |5| + |11|)\, n^3$$

$$\le 39\, n^3$$

$\therefore f(n)$ is $O(n^3)$.

Def Let f be a function from $\mathbb{N}$ to $\mathbb{N}$.
We say that $f(n)$ is $O(g(n))$
if there exist positive integers k and n_0
such that

$$f(n) \le k\, g(n)$$

for all $n \ge n_0$.

Remark The symbol $O(g(n))$ is pronounced
"big-oh of $g(n)$."

Example

$$\begin{aligned} f(n) &= 3n^3 - 20n^2 + 5n - 11 \\ &\le (|3| + |20| + |5| + |11|)\, n^3 \\ &\le 39\, n^3 \end{aligned}$$

$\therefore f(n)$ is $O(n^3)$.

Example $n^2 + 17n + 3$ is $O(n^2)$.

$2^n + 3n^5 + 12n^4$ is $O(2^n)$.

Example Sorting algorithms

A sequence of distinct numbers $x_1, x_2, \cdots, x_n$
(in general, a totally ordered set) is
given and it is desired to arrange

A sequence of distinct numbers $x_1, x_2, \cdots, x_n$ (in general, a totally ordered set) is given and it is desired to arrange them in the increasing order, i.e., to have a permutation of the set $\{1, 2, \cdots, n\}$ for which

$$x_{\pi(1)} < x_{\pi(2)} < x_{\pi(3)} < \cdots < x_{\pi(n)}.$$

<u>Example</u> $n^2 + 17n + 3$ is $O(n^2)$.

$2^n + 3n^5 + 12n^4$ is $O(2^n)$.

<u>Example</u> Sorting algorithms

A sequence of distinct numbers $x_1, x_2, \cdots, x_n$ (in general, a totally ordered set) is given and it is desired to arrange them in the increasing order, i.e., to have a permutation of the set $\{1, 2, \cdots, n\}$ for which

$$x_{\pi(1)} < x_{\pi(2)} < x_{\pi(3)} < \cdots < x_{\pi(n)}.$$

"Bubble sort"

for $j := 1$ to $n-1$

for $i := 1$ to $n-j$

if $x_i > x_{i+1}$ then interchange x_i and x_{i+1}

if $x_i > x_{i+1}$ then interchange x_i and x_{i+1}

At the jth pass, $n-j$ comparisons are made,

and so total # of comparisons

$$= (n-1) + (n-2) + \cdots + 2 + 1$$

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

which is $O(n^2)$.

"Bubble sort"

for $j := 1$ to $n-1$

for $i := 1$ to $n-j$

if $x_i > x_{i+1}$ then interchange x_i and x_{i+1}

At the jth pass, $n-j$ comparisons are made,

and so total # of comparisons

$$= (n-1)+(n-2)+\cdots+2+1$$

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

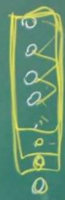
which is $O(n^2)$

"Bubble sort"

for $j := 1$ to $n-1$

for $i := 1$ to $n-j$

if $x_i > x_{i+1}$ then interchange x_i and x_{i+1}



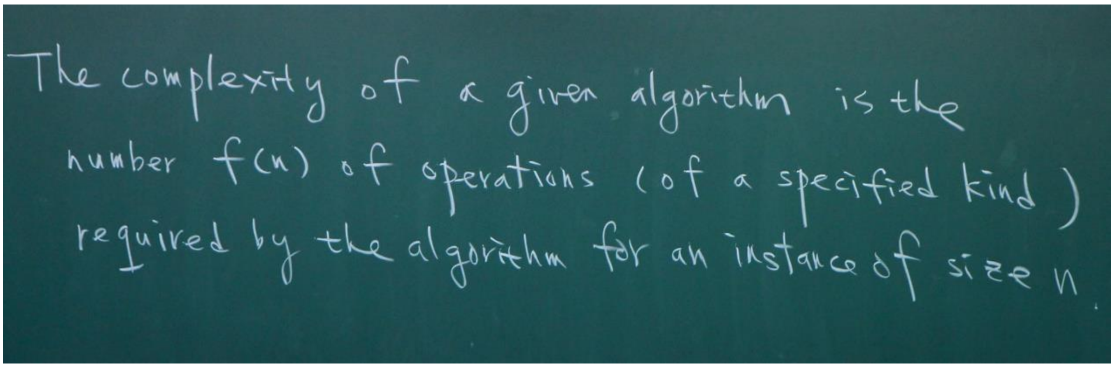
At the jth pass, $n-j$ comparisons are made,

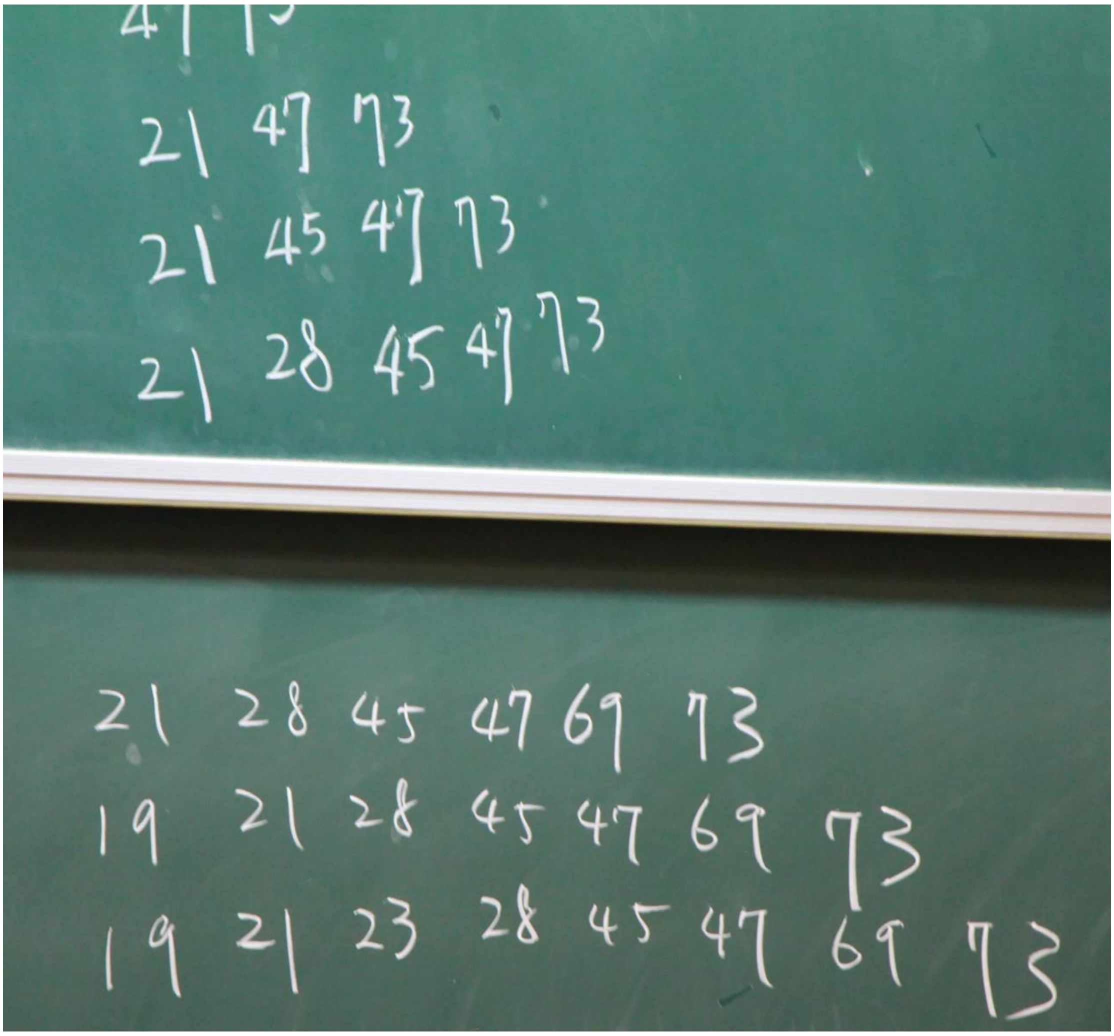
and so total # of comparisons

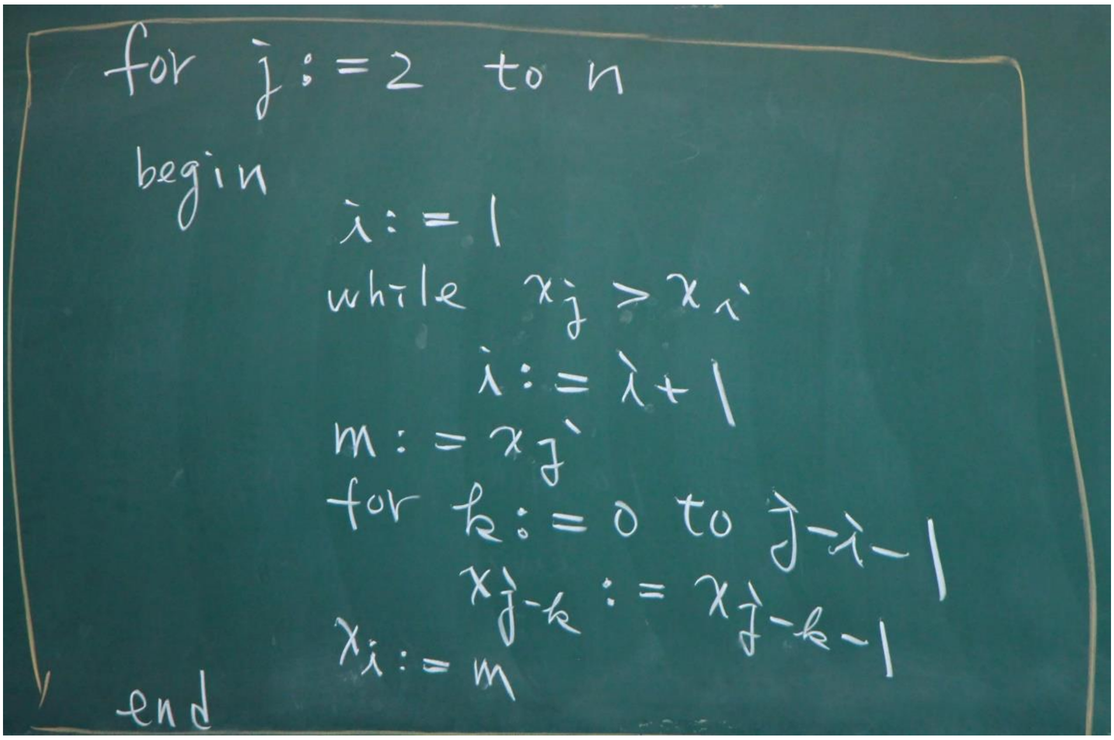
$$= (n-1)+(n-2)+\cdots+2+1$$

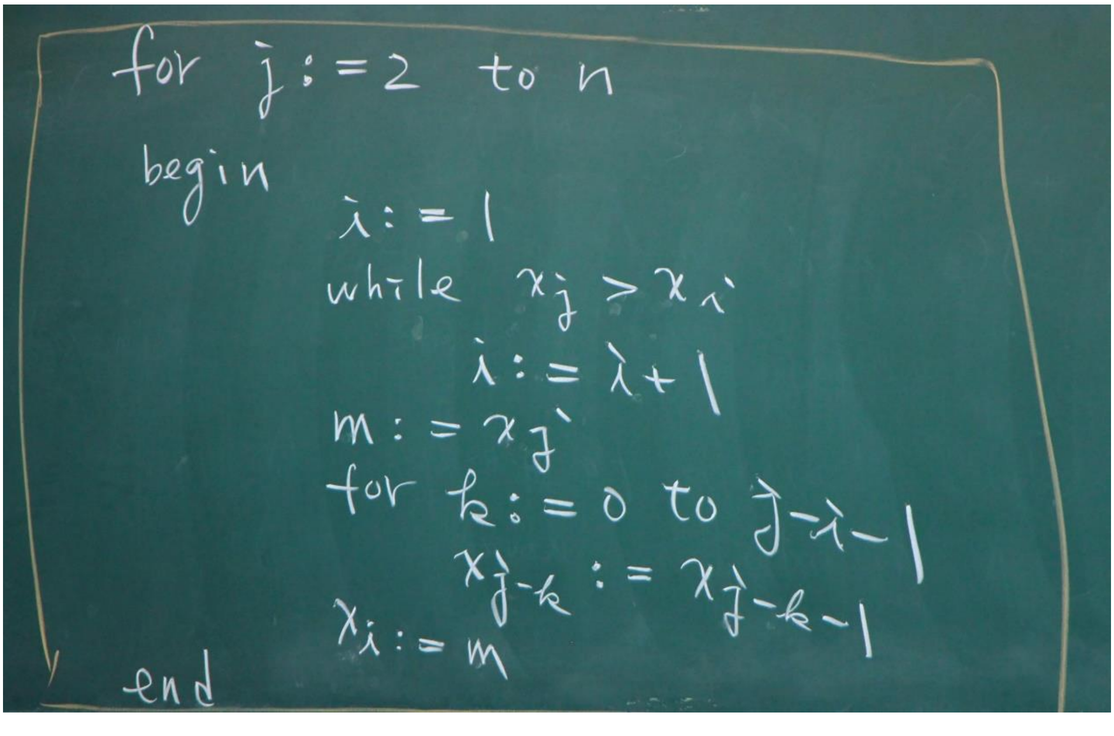
$$= \frac{n(n-1)}{2}$$

which is $O(n^2)$.


The complexity of a given algorithm is the
number f(n) of operations (of a specified kind)
required by the algorithm for an instance of size n.


21 47 73
21 45 47 73
21 28 45 47 73
21 28 45 47 69 73
19 21 28 45 47 69 73
19 21 23 28 45 47 69 73


for j:=2 to n
begin
i:=1
while $x_j > x_i$
i:=i+1
m:=x_j
for k:=0 to j-i-1
$x_{j-k} := x_{j-k-1}$
x_i:=m
end


for j:=2 to n
begin
i:=1
while $x_j > x_i$
i:=i+1
m:=x_j
for k:=0 to j-i-1
$x_{j-k} := x_{j-k-1}$
x_i:=m
end

```
begin
    i := 1
    while x_j > x_i
        i := i + 1
    m := x_j
    for k := 0 to j-i-1
        x_{j-k} := x_{j-k-1}
    x_i := m
end
```

At the jth pass, we could need as many as $(j-1)$ comparisons of x_j. So

total # of comparisons $= 1+2+3+\cdots+(n-1)$

(worst-case) $= \dfrac{n(n-1)}{2}$

which is $O(n^2)$.

There are sorting algorithms which can achieve $O(n\log n)$ comparisons (in the worst case).

```
for j := 2 to n
```


```
begin
    i := 1
    while x_j > x_i
        i := i + 1
    m := x_j
    for k := 0 to j-i-1
        x_{j-k} := x_{j-k-1}
    x_i := m
end
```

At the jth pass, we could need as many as $(j-1)$ comparisons of x_j. So

total # of comparisons $= 1+2+3+\cdots+(n-1)$

(worst-case) $= \dfrac{n(n-1)}{2}$

which is $O(n^2)$.

There are sorting algorithms which can achieve $O(n\log n)$ comparisons (in the worst case).