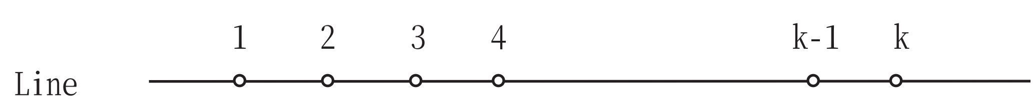
4 Solved By Combinatorial Argument

In this section, we will solve these three questions directly by Combinatorial Argument. For convenience, we have the following definitions: A ***k-max-point-drawing*** to be a drawing with k different points in a line such that these k points create the largest number of different regions in this line, and two k-max-point-drawings are isomorphic if their underlying graphs of these k points are isomorphic, otherwise they are non-isomorphic. A ***k-max-line-drawing*** to be a drawing with k different lines in a plane such that these k lines create the largest number of different regions in this plane, and two k-max-line-drawings are isomorphic if their underlying graphs of the intersection points are isomorphic, otherwise they are said to be non-isomorphic. A ***k-max-plane-drawing*** to be a drawing with k different planes in a 3-dimensional space such that these k planes create the largest number of different regions in this space, and two k-max-plane-drawings are isomorphic if their underlying graphs of the intersection points are isomorphic, otherwise they are non-isomorphic.

4.1 Non-isomorphic of k-Max-Line-Drawing and k-Max-Plane-Drawing

Lemma 4.1 *All k-max-point-drawings are isomorphic.*

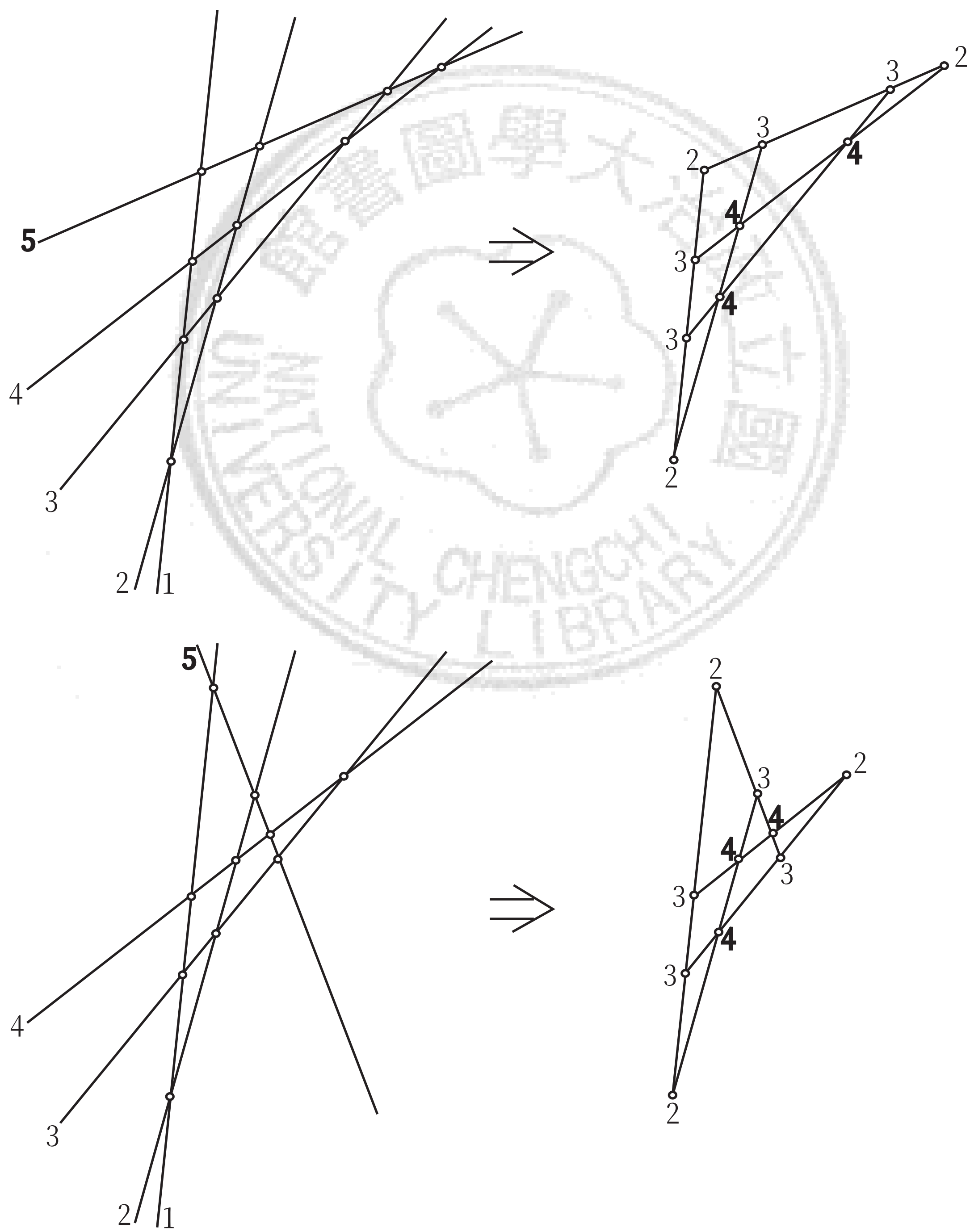
Proof. We can easily see that all k-max-point-drawings are isomorphic in the following figure.



❑

Lemma 4.2 *There exist two k-max-line-drawings are non-isomorphic.*

Proof. The underlying graphs are non-isomorphic since the three vertices of degree 4 forms an K_3 in the upper one and an P_3 in the lower one. Hence, we easily see that the two 5-max-line-drawings are non-isomorphic.



❑

Lemma 4.3 *There exist two k-max-plane-drawings are non-isomorphic.*

***Proof*.** We know that choosing any two of k different planes will intersect in a common line, and there exist two non-isomorphic k-max-line-drawing. Therefore, we conclude that there exist two non-isomorphic k-max-plane-drawing. ❑

4.2 Combinatorial Argument with Algorithm

Since the existence of non-isomorphic k-max-line-drawing and k-max-plane-drawing, we can't provide a special drawing of k lines or k planes to show the formula by combinatorial argument. Here we use Algorithms to label all regions each by levels, and collect all the labels to explain the formula directly.

Algorithm 4.4 *(Label of k-max-point-drawing)*
Input: k points and a line such that these k points form the greatest number of regions in the line.
Initialization: Set $L_1 = \{0\}$
Iteration: Label each regions formed by these k points as follows.

1. Label POINT_p as p.
2. Do step 1 for $p = 1 \sim k$.
3. Label each region on this line as the label of its point from one end of this line to another end.

Output: Label of *k-max-point-drawing.*

***Proof*.** The output of this algorithm is $L_1 = \{0, 1, 2, 3, \ldots, k\}$.
Hence we can say that, $\{0\}$ stands for C_0^k, and $\{1, 2, 3, \ldots, k\}$ stands for C_1^k. This completes the proof of the largest number in a line made of k different points is just $C_0^k + C_1^k$. ❑

Algorithm 4.5 *(Label of k-max-line-drawing)*
Input: k lines and a plane such that these k lines forms the greatest number of

regions in the plane.

Initialization: Set $L_2 = \{(0, 0)\}$

Iteration: Label each regions formed by these k lines as follows.

1. Label the intersection points made up by the p^{th} line with each of the former lines $\text{LINE}_1 \sim \text{LINE}_{p-1}$ as $\{1, 2, 3, \ldots, p-1\}$.

2. Label each of the p^{th} line's regions in the upper plane of the p^{th} line as $L_1 = \{0, 1, 2, \ldots, p-1\}$ by Algorithm 4.4.

3. Label each of the increasing regions made up by the p^{th} line associated with each region on the p^{th} line by $L_{2,p} = L_1 \times \{p\}$.

4. Set $L_2 = L_2 \cup L_{2,p}$.

5. Do step 1∼4 for $p = 1 \sim k$.

Output: Label of *k-max-line-drawing.*

Proof. The output of this algorithm is $L_2 = \{(0, 0), (0, 1), (0, 2), \ldots, (0, k), (1, 2), (1, 3), \ldots, (1, k), (2, 3), (2, 4), \ldots, (2, k), (3, 4), \ldots, (k-1, k)\}$.
Hence $\{(0, 0)\}$ stands for C_0^k, $\{(0, 1), (0, 2), \ldots, (0, k)\}$ stands for C_1^k, and $\{(1, 2), (1, 3), \ldots, (1, k), (2, 3), (2, 4), \ldots, (2, k), (3, 4), \ldots, (k-1, k)\}$ plays the role of C_2^k. This completes the proof of the greatest number of the regions made up by k different lines is $C_0^k + C_1^k + C_2^k$. ❑

Algorithm 4.6 *(Label of k-max-plane-drawing)*

Input: k planes in a space such that these k planes forms the greatest number of regions in the space.

Initialization: Set $L_3 = \{(0, 0, 0)\}$

Iteration: Label each regions formed by these k planes as follows.

1. Label the intersection lines made up by the p^{th} plane with each of the former planes $\text{PLANE}_1 \sim \text{PLANE}_{p-1}$ as $\{1, 2, 3, \ldots, p-1\}$.

2. Label each of the p^{th} plane's regions in the upper space of the p^{th} plane as $L_2 = \{(0, 0), (0, 1), (0, 2), \ldots, (0, p), (1, 2), (1, 3), \ldots, (1, p), (2, 3), (2, 4), \ldots, (2, p), (3, 4), \ldots, (p-1, p)\}$ by Algorithm 4.5.

3. Label each of the increasing regions made up by the p^{th} plane associated

with each region on the p^{th} plane by $L_{3,p} = L_2 \times \{p\}$.

4. Set $L_3 = L_3 \cup L_{3,p}$.
5. Do step $1 \sim 4$ for $p = 1 \sim k$.

Output: Label of *k-max-plane-drawing.*

Proof. The output of this algorithm is $L_2 = \{(0, 0, 0), (0, 0, 1), (0, 0, 2), \ldots, (0, 0, k), (0, 1, 2), (0, 1, 3), \ldots, (0, 1, k), (0, 2, 3), \ldots, (0, k-1, k), (1, 2, 3), (1, 2, 4), \ldots, (1, k-1, k), (2, 3, 4), (2, 3, 5), \ldots, (2, k-1, k), (3, 4, 5), \ldots, (k-2, k-1, k)\}$.
Hence $\{(0, 0, 0)\}$ stands for C_0^k, $\{(0, 0, 1), (0, 0, 2), \ldots, (0, 0, k)\}$ stands for C_1^k, $\{(0, 1, 2), (0, 1, 3), \ldots, (0, 1, k), (0, 2, 3), \ldots, (0, k-1, k)\}$ plays the role of C_2^k, and $\{(1, 2, 3), (1, 2, 4), \ldots, (1, k-1, k), (2, 3, 4), (2, 3, 5), \ldots, (2, k-1, k), (3, 4, 5), \ldots, (k-2, k-1, k)\}$ plays the role of C_3^k. This completes the proof of the greatest number of the regions made up by k different lines is $C_0^k + C_1^k + C_2^k + C_3^k$.
❑

4.3 Combinatorial Argument for Higher Dimensional Space with Algorithm

For higher dimensional spaces, we use the similar Algorithms to label all regions. We also easily see that the labels increase an coordinate for each increasing dimension. Hence, this not only successfully provide a combinatorial proof of the formula, but also explains each component of the formula.

That is to say, if you know how to label the regions in $(n-1)$-dimensional space, you will know how to label each region in n-dimensional space by adding an coordinate which is obtained from cartesian product with the p^{th} hyperplane in the Algorithm.

4.4 Presentation of Partitions in the Lower Dimensional Space

First, we set $n = 1$ and use familiar notations as $x \equiv x_1$ in the *Standard Partition System of n-Dimensional Space* to show the presentation of partitions in 1-dimensional space(i.e. *x-line*).

$$\begin{aligned}
m &= 1 &:& \quad x = 0 \\
m &= 2 &:& \quad x = 1 \\
m &= 3 &:& \quad x = 2 \\
&\vdots & & \quad \vdots \\
m &= k-1 &:& \quad x = k-2 \\
m &= k &:& \quad x = k-1
\end{aligned}$$

Then we could draw these k points in a x-line as Figure 5 (here we draw 7 points for convenience). Now we can explain why the formula ends up with C_1^k. As we use a big enough box to contain all the points on the line which is drawn obliquely, we gravel a crowd of sand to each region on the line equally. Then these sand will be pilled up at each point, and we can represents each regions by these crowd of sand.

So we can say that the formula is essentially added from C_0^k to C_1^k since choosing none of these k points provides a region which is the original $x - line$, and choosing any one of these k points will make a different region from the others on this line. Hence we don't need C_2^k or more other combinations since each region is supported by just only one point and the others are ineffective.

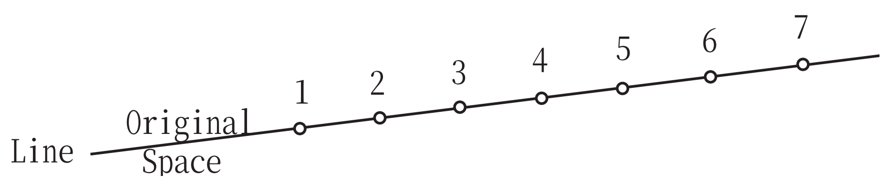


Figure 5: Presentation of Standard Partition System of 1-Dimensional Space

Second, we set $n = 2$ and use familiar notations as $y \equiv x_2$ and $x \equiv x_1$ in the *Standard Partition System of n-Dimensional Space* to show the presentation of partitions in 2-dimensional space(i.e. *xy-plane*).

$$\begin{array}{lll} m = 1 & : & y = x \\ m = 2 & : & y = \frac{1}{2}x + 1 \\ m = 3 & : & y = \frac{1}{3}x + 2 \\ \vdots & & \vdots \\ m = k-1 & : & y = \frac{1}{k-1}x + (k-2) \\ m = k & : & y = \frac{1}{k}x + (k-1) \end{array}$$

Then we could draw these k lines in a xy-plane as Figure 6 (here we draw 7 lines for convenience). Now we can realize why the formula ends up with C_2^k by the same way as before. As we use a big enough box to contain all the intersection points about these k lines in the $xy-plane$, we have all the regions made up by these k lines within this box. Then we gravel a crowd of sand to each region in this box equally, and this act makes that each crowd of sand is pilled up at each line, intersection point, and the bottom of this box. Hence these crowd of sand can be used to represent every region on the $xy-plane$ made up by these k lines.

Again, it is clearly that the formula adds form C_0^k to C_2^k since choosing none from these k lines provides an original $xy-plane$, and choosing any one of these k lines makes an individual region, and choosing any two of these k lines makes a different region from others.

Also, the formula doesn't need C_3^k or other combinations since each region is supported by only one or two lines and the third or the fourth line are ineffective.

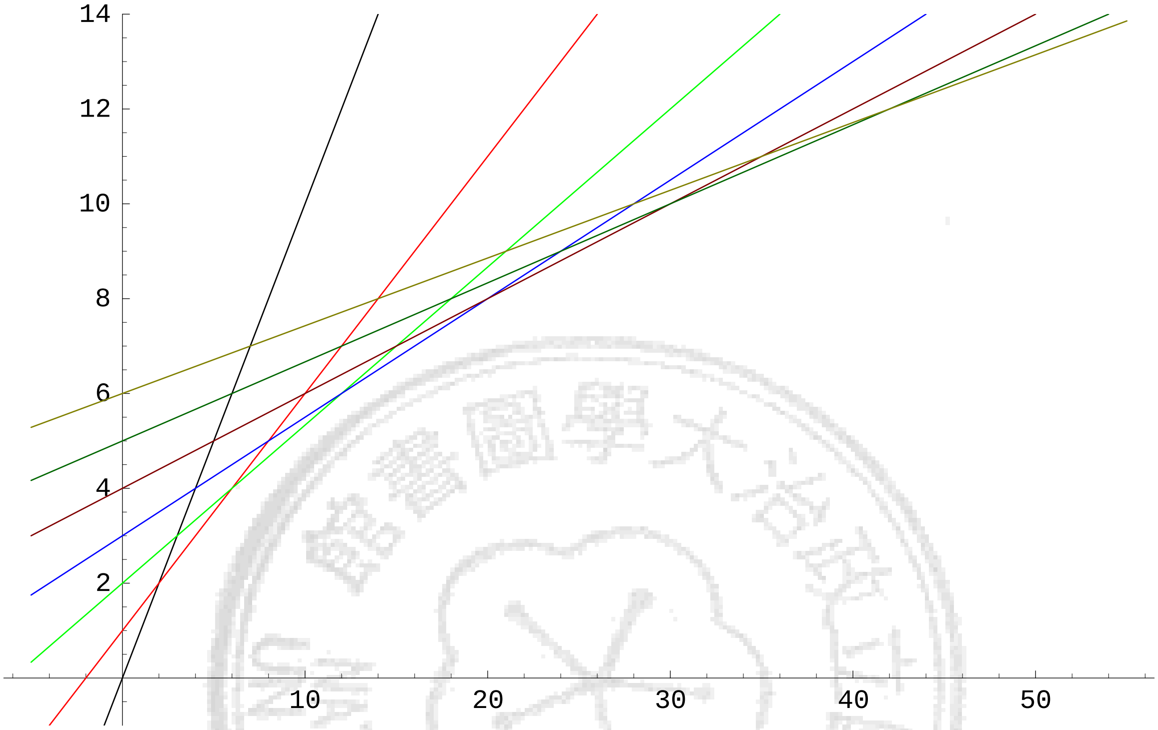


Figure 6: Presentation of Standard Partition System of 2-Dimensional Space

At last, we set $n = 3$ and use familiar notations as $z \equiv x_3$, $x \equiv x_1$, and $y \equiv x_2$ in the *Standard Partition System of n-Dimensional Space* to show the representation of partitions in 3-dimensional space(i.e. *xyz-space*).

$$
\begin{array}{lcl}
m = 1 & : & z = x + \frac{1}{2}y \\
m = 2 & : & z = \frac{1}{2}x + \frac{1}{3}y + 1 \\
m = 3 & : & z = \frac{1}{3}x + \frac{1}{4}y + 2 \\
\quad\vdots & & \vdots \\
m = k-1 & : & z = \frac{1}{k-1}x + \frac{1}{k}y + (k-2) \\
m = k & : & z = \frac{1}{k}x + \frac{1}{k+1}y + (k-1)
\end{array}
$$

Then we could draw these k lines in a xyz-space as Figure 7 (here we draw 5 planes for convenience). Now we can realize why the formula adds from C_0^k and ends up with C_3^k. Since each region is supported by just only one plane , two planes, or three planes. And all the others are ineffective. The representation is just the same way as in $x - line$ and in $xy - plane$.

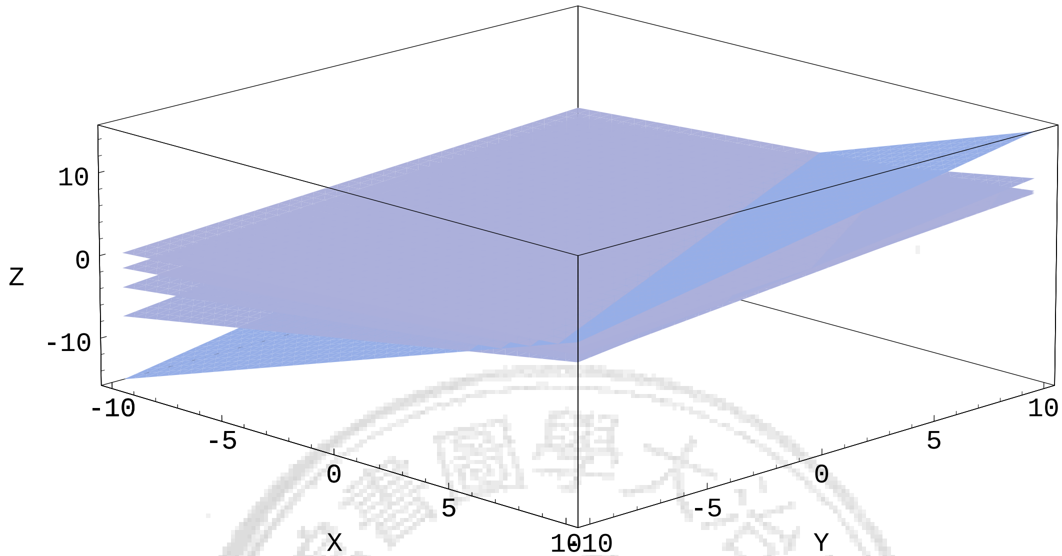


Figure 7: Presentation of Standard Partition System of 3-Dimensional Space

4.5 A List of All Numbers

We have the following list since the form of recurrence relation $P_{n,k} = P_{n,k-1} + P_{n-1,k-1}$ in Section 3.5.

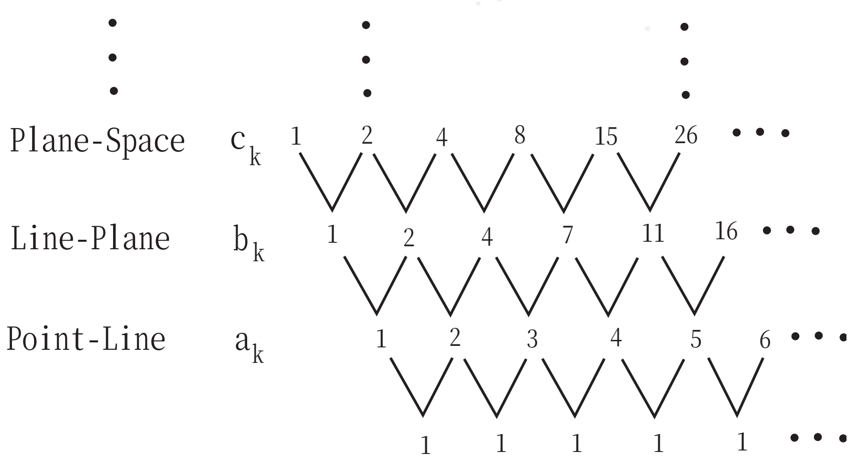


Figure 8: A List of All Numbers