# Recursion in Combinatorics

Hein Thant Aung

December 29, 2023

## 1 Introduction

ပုစ္ဆာအကြီးတစ်ခုကိုဖြေရှင်းဖို့အတွက် ဒီပုစ္ဆာအကြီးကို ပုံစံတူပုစ္ဆာအသေးလေးတွေဖြစ်အောင် ပြောင်းပစ်တဲ့ idea ကို mathematical induction မှာတုန်းက တွေ့ဖူးပြီးသားဖြစ်မှာပါ။ ဒီ idea ကို combinatorics မှာလည်းသုံးလို့ရပါတယ်။ ဒီ handout ရဲ့ section 2 နဲ့ 3 ကတော့ "ဒီ idea ကို counting ပုစ္ဆာတွေမှာ ဘယ်လိုအသုံးချမလဲ" ဆိုတဲ့ အသုံးချပုံကိုပြထားတာပါ။ တကယ်စိတ်ဝင်စားဖို့ကောင်းတဲ့ အသုံးချပုံတွေကိုတော့ section 4: Recursive Algorithms မှာကြည့်နိုင်ပါတယ်။ ဒီ handout ကိုဖတ်ဖို့ အခြေခံ counting principle တွေဖြစ်တဲ့ ပေါင်းခြင်းနဲ့ မြှောက်ခြင်းအကြောင်းသိရင်ရပါပြီ။ Permutations and combinations အကြောင်းက သိထားရင်ကောင်းပေမယ့် သိစရာမလိုပါဘူး။

## 2 Recursion in Counting

Recursion ဆိုတာဘာလဲမပြောခင် ပုစ္ဆာတချို့နဲ့စကြည့်ကြည့်ရအောင်။

**Example 2.1.** How many ways are there are to tile a $1 \times 10$ rectangles using identical $1 \times 1$ rectangles and identical $1 \times 2$ rectangles?

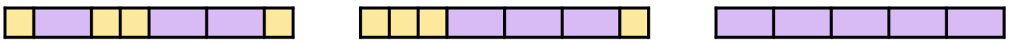


Figure 1: ဥပမာ tiling တချို့

**Solution.** အရင်ဆုံး $1 \times 10$ rectangle ကို tile လုပ်မယ့်အစား $1 \times n$ rectangle လေးတွေကို $n$ တန်ဖိုးအငယ်လေးတွေအတွက် အရင် tile ကြည့်ရအောင်။ $n = 1, 2, 3, 4, 5$ အတွက် ဖြစ်နိုင်တဲ့ tiling အကုန်ချဆွဲကြည့်ရင် Figure 1 အတိုင်းရပါမယ်။ အရေအတွက်တွေက $1, 2, 3, 5, 8$ ဆိုပြီးဖြစ်နေပါတယ်။

**Quick Exercise 2.2.** Find a pattern in the sequence $1, 2, 3, 5, 8$ and confirm your answer by listing all tilings of the $1 \times 6$ rectangle.

ဒီ sequence $1, 2, 3, 5, 8$ ကိုကြည့်ကြည့်ရင် "ရှေ့ကိန်းသည် နောက်ကိန်းနှစ်လုံး၏ပေါင်းလဒ်" ဆိုတဲ့ pattern ကို တွေ့နိုင်ပါတယ်။ ဒါကြောင့် $1 \times n$ ကို tile နိုင်တဲ့နည်းအရေအတွက်ကို $f_n$ လို့ထားရင်

$$f_n = f_{n-1} + f_{n-2}, \quad \text{for } n \geq 3 \qquad (\star)$$

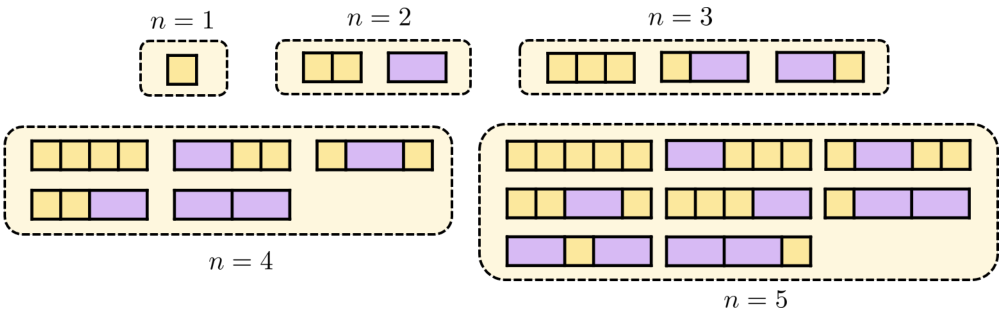



Figure 2: $n = 1, 2, 3, 4, 5$ အတွက် $1 \times n$ rectangle ကို tile နိုင်တဲ့နည်းတွေက $1, 2, 3, 5, 8$ ဆိုပြီးဖြစ်နေတယ်

ဖြစ်သင့်တာပေါ့။ ဒီ $(\star)$ ကိုမှန်ကြောင်းသာပြနိုင်ရင် $f_{10}$ ကိုရှာတာက လွယ်လွယ်လေးဖြစ်သွားပါပြီ။ ဒါဖြင့်ရင် $(\star)$ က ဘာလို့မှန်တာလဲ။ $1 \times n$ rectangle ကို tile လုပ်ဖို့အတွက် ဘယ်ဘက်စွန်ဆုံးက rectangle သည် $1 \times 1$ ဖြစ်ရင်ဖြစ် ဒါမှမဟုတ် $1 \times 2$ rectangle ဖြစ်မှာပါ။ ဒါကြောင့်

$$f_n = \text{number of tilings starting with } 1 \times 1 \text{ rectangle} + \text{number of tilings starting with } 1 \times 2 \text{ rectangle} \quad (\star\star)$$

ဖြစ်ပါတယ်။ ဒါပေမယ့် $n \geq 3$ အတွက် $1 \times 1$ rectangle နဲ့စတဲ့ tiling အရေအတွက်က tile ဖို့ကျန်နေတဲ့ $1 \times (n-1)$ rectangle ကို tile နိုင်တဲ့နည်းအရေအတွက်ပဲဖြစ်လို့ $(\star\star)$ ညာဘက်ခြမ်းရဲ့ ပထမ term သည် $f_{n-1}$ နဲ့အတူတူပဲဖြစ်ပါတယ်။ ထိုနည်းတူစွာပဲ $(\star\star)$ ရဲ့ ဒုတိယ term သည် $f_{n-2}$ နဲ့အတူတူပဲဖြစ်ပါတယ်။ ဒါကြောင့် ညီမျှခြင်း $(\star)$ က $n \geq 3$ တိုင်းအတွက် မှန်နေတာပေါ့။ ဒါကြောင့် $f_{10}$ ကို ဒီလိုတွက်လိုက်လို့ရပါပြီ။

$$f_6 = 8 + 5 = 13, \quad f_7 = 13 + 8 = 21, \quad f_8 = 21 + 13 = 34, \quad f_9 = 34 + 21 = 55, \quad f_{10} = 55 + 34 = 89$$

ဒါကြောင့် အဖြေက $89$ ဖြစ်ပါတယ်။ □

---

**Example 2.3.** How many ways are there to arrange $10$ coins in a row so that there are no two consecutive heads (see figure 3)?

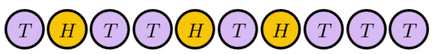



Figure 3: ဥပမာ အစီအစဉ်တစ်ခု

**Solution.** $10$ နေရာမှာ $n$ လဲထည့်ပြီး စီနိုင်တဲ့နည်းအရေအတွက်ကို $g_n$ လို့ထားလိုက်ပါ။ ပြီးခဲ့တဲ့ example ကအတိုင်းပဲ $g_n$ အတွက် pattern ရှာကြည့်ရအောင်။ ဒင်္ဂါး $n$ ခုကို အတန်းလိုက်စီထားမယ်ဆိုရင် ရှေ့ဆုံးဒင်္ဂါးက $T$ သို့မဟုတ် $H$ နှစ်မျိုးပဲဖြစ်နိုင်တယ်။ ဒါကြောင့်

$$g_n = \text{number of arrangements starting with } T + \text{number of arrangements starting with } H \quad (\star)$$

ဖြစ်တယ်။ ဒါပေမယ့် $T$ နဲ့စရင် နောက်လာမယ့် ဒင်္ဂါး $n-1$ ပြားက ခေါင်းနှစ်ခုဆက်တိုက်မပါရင်ပြီးရော၊ ဖြစ်ချင်တာဖြစ်လို့ရတယ်။ ဒါကြောင့် $(\star)$ ညာဘက်ခြမ်းရဲ့ ပထမ term သည် $g_{n-1}$ နဲ့အတူတူပဲ။ ရှေ့ဆုံးဒင်္ဂါးက $H$ ဖြစ်ရင်ကျတော့ သူ

နဲ့ကပ်လျက်ဒဂ်ါးက $T$ ဖြစ်ကိုဖြစ်ရမယ်။ ဒီ $T$ ဘယ်ဘက်က ဒဂ်ါး $n-2$ ပြားကျတော့ ခေါင်းနှစ်ခုဆက်တိုက်မပါရင်ပြီး ရော၊ ဖြစ်ချင်တာဖြစ်လို့ရတယ်။ ဒါကြောင့် $(\star)$ ညာဘက်ခြမ်းရဲ့ ဒုတိယ term သည် $g_{n-2}$ နဲ့အတူတူပဲ။ အခုလုပ်သွားတဲ့ argument သည် $n \geq 3$ ဖြစ်မှလုပ်လို့ရပါတယ် (why?)။ ဒါကြောင့်

$$g_n = g_{n-1} + g_{n-2}, \quad \text{for all } n \geq 3$$

ဆိုတဲ့ ဆက်သွယ်ချက်ကိုရသွားပါပြီ။ ဒါဆိုရင် $g_1$ နဲ့ $g_2$ ကိုရှာပြီးတော့ ကျန်တာက လက်နဲ့ဆက်တွက်လိုက်ရုံပါပဲ။ $g_1 = 2$, $g_2 = 3$, $g_3 = 5$, $g_4 = 8, \ldots, g_{10} = 144$ ရပါတယ်။ $\square$

---

**Remark.** ပြီးခဲ့တဲ့ example နှစ်ပုဒ်အရ $n \geq 2$ အတွက် $f_n = g_{n-1}$ ဖြစ်ကြောင်း တွေ့နိုင်ပါတယ်။ ဘာလို့မှန်သလဲဆိုတာကို အကြမ်းပြောရရင်တော့ $f_2 = g_1$ နဲ့ $f_3 = g_2$ ဖြစ်ပြီး ကိန်းစဉ်နှစ်ခုလုံးက "ရှေ့ကိန်းသည် နောက်ကိန်းနှစ်လုံးပေါင်းလဒ်" ဆိုတဲ့ pattern ကိုလိုက်နေလို့ပါပဲ။ ဒီ idea ကိုတိတိကျကျချရေးချင်ရင်တော့ (strong) induction နဲ့ရေးလို့ရပါတယ်။

**Remark.** ဒီ $f_n = g_{n-1}$ ဖြစ်ရတဲ့အကြောင်းရင်းကို recurrence relation (pattern) မရှာဘဲ တိုက်ရိုက်ပြလို့ရပါတယ်။ သက်သေပြချက်ကို စာနဲ့မရေးတော့ပါဘူး။ Figure 4 ကိုကြည့်ပြီး ကိုယ့်ဘာသာစဉ်းစားကြည့်ပါ။

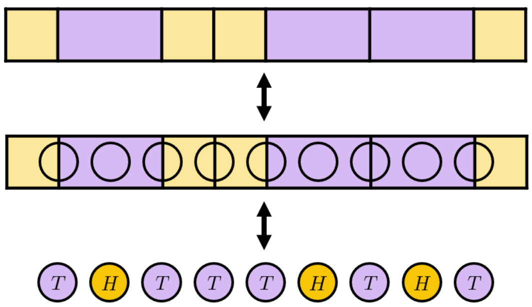



Figure 4: Proof of $f_n = g_{n-1}$

**Example 2.4.** Let $n$ be a positive integer. A total of $n$ lines are drawn on the plane so that no two are parallel and no three are concurrent. What is the number of regions do these lines divide the plane into?

**Solution.** အဖြေကို $a_n$ လို့ထားလိုက်ပါ (အခုချိန်မှာ $a_n$ က $n$ ပေါ်မှာပဲမှီခိုသလား၊ မျဉ်းတွေကိုဘယ်လိုဆွဲသလဲ ဆိုတာပေါ်ရော မှီခိုသေးသလားဆိုတာ မသိနိုင်သေးပါဘူး။ ဒီ solution က $a_n$ သည် $n$ ပေါ်မှာပဲမှီခိုကြောင်းပါ တခါတည်းပြသွားပါလိမ့်မယ်)။ အရင်ဆုံး $a_n$ နဲ့ သူ့ရှေ့က term တွေ ဘယ်လိုချိတ်ဆက်သလဲဆိုတာ ကြည့်ရအောင်။ မျဉ်း $n$ ကြောင်း ပါတဲ့ configuration တစ်ခုယူလိုက်ပါ။

ဒီ ထဲ က တစ် ကြောင်း ကို ဖယ် လိုက် ရင် မျဉ်း $n-1$ ကြောင်း ပါ တဲ့ configuration ဖြစ် သွား တဲ့ အတွက် region အရေအတွက်က $a_{n-1}$ ဖြစ်သွားပါတယ်။ ပြီးတော့မှ စောနကဖယ်လိုက်တဲ့ မျဉ်းကိုပြန်ဆွဲလိုက်ပါ။ ပြန်ဆွဲရင်းနဲ့ မူရင်း $n-1$ ထဲကတစ်ကြောင်းကိုထိတိုင်း region အရေအတွက်က တစ်ခုတိုးလာတယ် (See figure 5)။ မျဉ်းတစ်ကြောင်း

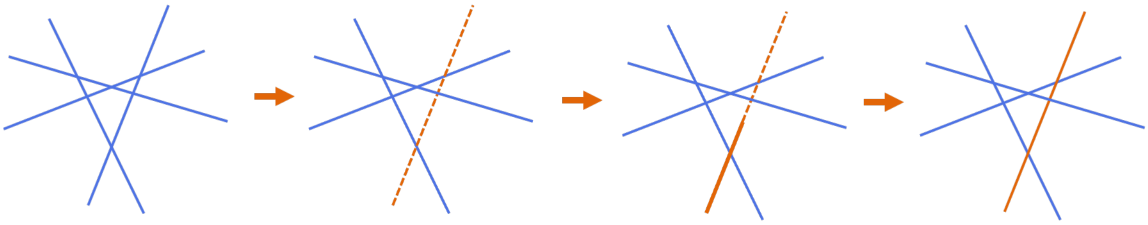


Figure 5: $a_n$ နဲ့ $a_{n-1}$ ဆက်သွယ်ချက်ကို ရှာပုံ

လုံး ဆွဲ ပြီး သွား တဲ့ အခါ လည်း region နောက်တစ်ခု တိုး လာ တယ်။ ဒါ ကြောင့် မျဉ်း $n$ ကြောင်း ပါ တဲ့ ပုံ မှာ ရှိ တဲ့ region အရေအတွက်သည် $a_{n-1} + (n-1) + 1$ ဖြစ်တယ်။ ဒါဆိုရင်

$$a_n = a_{n-1} + n, \quad \text{for } n \geq 2$$

ဆို တဲ့ ဆက် သွယ် ချက် ရ သွား ပါ ပြီ။ ဒါ ဆို ရင် $a_n$ ကို in terms of $n$ နဲ့ ပဲ ရှာ တာ က ကျောင်း စာ အဆင့် ဖြစ် သွား ပါ ပြီ။ ပြောင်းပြန် substitute သွားရုံပါပဲ။

$$\begin{aligned}
a_n &= a_{n-1} + n \\
&= a_{n-2} + (n-1) + n \\
&= a_{n-3} + (n-2) + (n-1) + n \\
&= \cdots \\
&= a_1 + 2 + 3 + \cdots + (n-1) + n \\
&= 1 + \frac{n(n+1)}{2}
\end{aligned}$$

□

---

**Quick Exercise 2.5.** Let $n$ be a positive integer. There are $n$ circles drawn on the plane so that any two circles intersect at $2$ points, and no three circles meet at a point. How many regions do these circles divide the plane into?

ဒီလိုမျိုး ကိန်းစဉ်တစ်ခု $a_1, a_2, a_3, \ldots$ ရှိပြီးတော့ ဒီကိန်းစဉ်ရဲ့ $a_n$ ကို သူ့ရှေ့က term တွေဖြစ်တဲ့ $a_{n-1}, a_{n-2}, \ldots, a_1$ တွေနဲ့ ဖော်ပြထားတဲ့ ညီမျှခြင်းကို recurrence relation လို့ခေါ်ပါတယ်။ ဥပမာအားဖြင့်

$$\begin{aligned}
a_n &= a_{n-1} + 4 && \text{for } n \geq 2 \\
a_n &= 2a_{n-1} && \text{for } n \geq 2 \\
a_n &= a_{n-1} + a_{n-2} && \text{for } n \geq 3 \\
a_n &= a_{n-1}a_1 + a_{n-2}a_2 + \cdots + a_1a_{n-1} && \text{for } n \geq 2
\end{aligned}$$

စတဲ့ဆက်သွယ်ချက်တွေဟာ recurrence relation တွေဖြစ်ကြပါတယ်။ ပထမညီမျှခြင်းက ကိန်းစဉ်သည် arithmetic progression ဖြစ်ကြောင်းကိုပြောတာနဲ့ တူတူပါပဲ (why?)။ ဒုတိယကိန်းစဉ်ကတော့ geometric progression ပါ (why?)။ တတိယကိန်းစဉ်ရဲ့ ဆက်သွယ်ချက်ကတော့ Fibonacci ဆက်သွယ်ချက်ပါ။ Example 2.1 နဲ့ 2.3 မှာ တွေ့ခဲ့တဲ့ ဆက်သွယ်ချက် ပါပဲ။ စတုတ္ထ ကိန်းစဉ်ကိုတော့ နောက်ကျရင် တွေ့ပါလိမ့်မယ်။ ကိန်းစဉ်တစ်ခုကို အတိအကျသတ်မှတ်ဖို့အတွက် recurrence relation ကိုပြောရုံနဲ့ မလုံလောက်ပါဘူး။ ရှေ့ဆုံးကိန်းတွေဖြစ်တဲ့ $a_1, a_2, \ldots$ စတဲ့ကိန်းတွေကို လုံလောက်သလောက် သတ်မှတ်ထားဖို့လိုပါတယ်။ ဥပမာ $1, 5, 9, 13, \ldots$ ဆိုတဲ့ကိန်းစဉ်ရော $0, 4, 8, 12, \ldots$ ဆိုတဲ့ကိန်းစဉ်ရောက $a_n = a_{n-1} + 4$ ဆိုတဲ့ recurrence relation ကိုပြေလည်ပါတယ်။ $a_1$ ရဲ့တန်ဖိုးကိုသိသွားမှသာ ဒီ recurrence relation ကိုသုံးပြီး ကိန်းစဉ်တစ်ခုလုံးကို အတိအကျ determine လုပ်လို့ရပါတယ်။ ဒီရှေ့ဆုံးကိန်းတွေရဲ့တန်ဖိုးတွေကို initial condition တွေလို့ခေါ်ပါတယ်။

**Quick Exercise 2.6.** Let $u_1, u_2, u_3, \ldots$ be a sequence defined by $u_1 = 3$ and

$$u_n = 2u_{n-1} - 4 \quad \text{for all } n \geq 2.$$

Find the formula for $u_n$.

## 3 Recursions with Pictures

Recursion တွေကို ကိန်းစဉ်တွေနဲ့ပဲလုပ်လို့ရတာမဟုတ်ပါဘူး။ ဘာကိုဆိုလိုသလဲသိဖို့ ဒီပုစ္ဆာကိုကြည့်ကြည့်ရအောင်။

**Example 3.7.** The map of Townsville is given in the figure 6. Josh wants to go from $A$ to $B$ by only walking northwards or eastwards.

(a) In how many ways can Josh go?

(b) If Josh is not allowed to go through $C$ and $D$, in how many ways can he go?

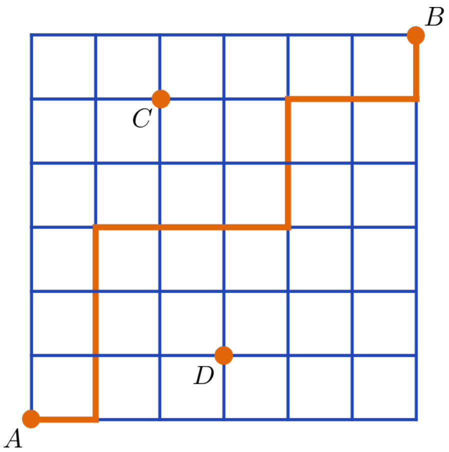



Figure 6: Townsville ရဲ့မြေပုံ

**Solution.** မြေပုံရဲ့ လမ်းဆုံလမ်းခွတစ်ခုချင်းစီမှာ $A$ ကနေပြီးတော့လာနိုင်တဲ့ နည်းလမ်းအရေအတွက်ကို ရေးလိုက်ပါ။ ဥပမာအားဖြင့် Figure 7 မှာပြထားသလိုပေါ့။

အောက်ဆုံးလမ်းနဲ့ ဘယ်ဘက်စွန်ဆုံးလမ်းပေါ်ကမဟုတ်တဲ့ လမ်းဆုံတစ်ခု $X$ မှာရေးရမယ့်ကိန်းကို recursively ရှာ

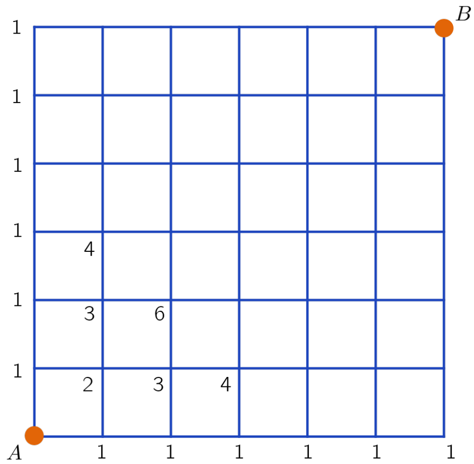



Figure 7: လမ်းဆုံတွေမှာ $A$ ကနေလာနိုင်တဲ့နည်းလမ်းအရေအတွက်ကို ရေးမယ်

လို့ရပါတယ်။ $X$ ရဲ့ဘယ်ဘက် (အနောက်ဘက်) ကပ်လျက်က လမ်းဆုံကို $Y$ လို့ခေါ်ပြီး အောက်ဘက် (တောင်ဘက်) ကပ်လျက်က လမ်းဆုံကို $Z$ လို့ခေါ်ရအောင်။ ဒါဆိုရင် $A$ ကနေ $X$ ကိုလာနိုင်တဲ့နည်းတွေကို အမျိုးအစားနှစ်မျိုးခွဲလို့ရပါတယ်။ $Y$ ကနေတစ်ဆင့်လာတဲ့နည်းတွေရယ်၊ $Z$ ကတစ်ဆင့်လာတဲ့နည်းတွေရယ်ဆိုပြီးတော့ပါ။ ဒါကြောင့်

$$\text{number at } X = \text{number at } Y + \text{number at } Z$$

ဖြစ်ပါတယ် (see figure 8)။

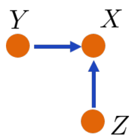



Figure 8: ရေးထားတဲ့ကိန်းတွေရဲ့ recurrence

ဒီဆက်သွယ်ချက်ကိုသုံးပြီးတော့ ရှိသမျှလမ်းဆုံတိုင်းမှာရှိတဲ့ကိန်းတွေကို Figure 9 ထဲကအတိုင်း တစ်ဆင့်တစ်ဆင့်ရှာလို့ရသွားပါပြီ။ ပုစ္ဆာရဲ့ part (b) သည်လည်း ဒီအတိုင်းပဲတွက်လို့ရပါတယ်။ လမ်းဆုံ $C$ နဲ့ လမ်းဆုံ $D$ ကို ပုံထဲကနေဖျက်လိုက်ရုံပါပဲ။

□

---

**Example 3.8.** Ten people sit around a round table. When a signal is given, some people immediately stand up. In how many ways can they stand so that no two adjacent people stand next to each other? (People are considered distinct from each other and so it is not necessary to consider rotations.)

**Solution.** လူတစ်ယောက် $A$ ကိုကြည့်လိုက်ပါ။ ထနည်းတွေထဲမှာ $A$ ကထိုင်လျက်သားနေတဲ့နည်းတွေနဲ့ $A$ ကမတ်တပ်ရပ်တဲ့နည်းတွေဆိုပြီး နှစ်မျိုးပဲရှိပါတယ်။ $A$ ကထိုင်လျက်သားနေနေတဲ့နည်းအရေအတွက်က ကျန်တဲ့လူ $9$ ယောက်ကို အတန်းလိုက်စီပြီး ကပ်လျက်နှစ်ယောက် တူတူမရပ်အောင် ထခိုင်းနိုင်တဲ့နည်းအရေအတွက်နဲ့ အတူတူပါပဲ။ $A$ ကမတ်တပ်ရပ်နေတဲ့နည်းအရေအတွက်ကတော့ ကျန်တဲ့လူ $7$ ယောက်ကို အတန်းလိုက်စီပြီး ကပ်လျက်နှစ်ယောက် တူတူမရပ်

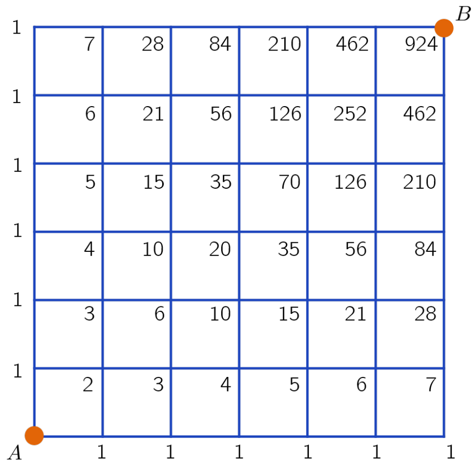



Figure 9: Recurrence သုံးပြီးအကုန်တွက်လိုက်လို့ရတယ်

အောင် ထခိုင်းနိုင်တဲ့နည်းအရေအတွက်နဲ့ အတူတူပါပဲ (see figure 10)။

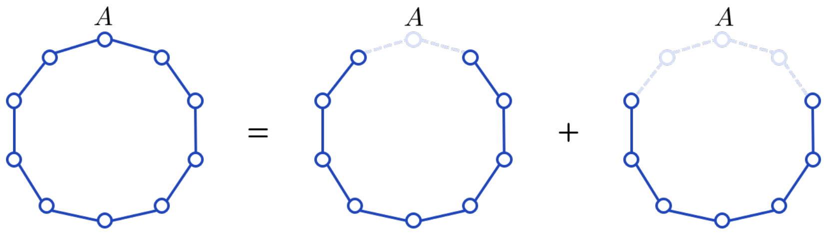



Figure 10: Recurrence ကို ပုံနဲ့မြင်လို့ရတယ်

ဒီနည်းတွေက Example 2.3 မှာတုန်းကတွေ့ခဲ့တဲ့ $g_9$ နဲ့ $g_7$ ပါပဲ။ ဒါကြောင့် အဖြေက $g_9 + g_7 = 89 + 34 = 123$ ဖြစ်ပါတယ်။ □

---

ဒီပုစ္ဆာကပေးတဲ့ recurrence က စက်ဝိုင်းပုံစီတာအတွက်ဆိုတာထက် အများကြီးပိုပြီး powerful ဖြစ်ပါတယ်။ User တချို့နဲ့ user အချင်းချင်းက friend or not friend ပဲဖြစ်နေနိုင်တဲ့ social network တစ်ခုကိုစဉ်းစားကြည့်လိုက်ပါ။ User တွေကို vertex လို့ခေါ်တဲ့ အမှတ်လေးတွေနဲ့ကိုယ်စားပြုပြီး အချင်းချင်း friend ဖြစ်နေတဲ့ user တွေကို edge လို့ခေါ်တဲ့အဆက်အသွယ်လေးတွေနဲ့ ချိတ်ပါမယ်။ ရလာတဲ့ vertex တွေရယ်၊ edge တွေရယ်ပါတဲ့ပုံကို graph (တစ်ခါတလေ network) လို့ခေါ်ပါတယ်။

ဒီ network မှာ user တချို့ရဲ့အစု $S$ (possibly empty) ရှိတယ် ဆိုပါစို့။ တကယ်လို့များ $S$ ထဲက user တွေက အချင်းချင်း friend မဟုတ်ရင်၊ တနည်းအားဖြင့် $S$ ထဲက vertex တွေကြားထဲမှာ ဘာ edge မှမရှိရင် $S$ ကို independent ဖြစ်တယ်လို့ခေါ်ပါတယ်။ ပုံ (figure 11) ထဲမှာ network တချို့ဥပမာပြထားပါတယ်။ အရောင်ခြယ်ထားတဲ့ user တွေရဲ့အစုတွေကတော့ independent set တွေအတွက် ဥပမာတွေဖြစ်ပါတယ်။

ဒါဆိုရင် example 3.8 က figure 11 ထဲကညာဘက်စွန်ဆုံးပုံမှာ independent set ဘယ်နှခုရှိသလဲဆိုတဲ့ မေးခွန်းနဲ့ အတူတူပါပဲ။ ဒီ example ထဲကနည်းအတိုင်းပဲ network တစ်ခုမှာ independent set ဘယ်နှခုပါသလဲဆိုတာကို ရှာလို့ရပါတယ်။

User တစ်ယောက် $u$ ကိုကြည့်လိုက်ပါ။ ဒါဆို independent set အရေအတွက်က $u$ ပါဝင်တဲ့ set အရေအတွက်နဲ့ $u$ မပါဝင်တဲ့ set အရေအတွက်ပေါင်းလဒ်နဲ့ အတူတူပါပဲ။ $u$ မပါဝင်တဲ့ set အရေအတွက်က network ထဲကနေ $u$ ကိုဖယ်ပြီး

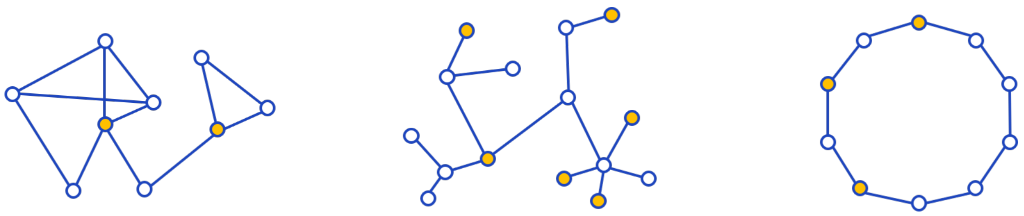


Figure 11: Network နဲ့ independent set ဥပမာအချို့

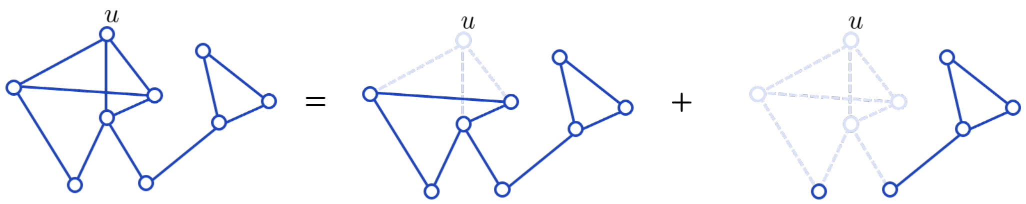
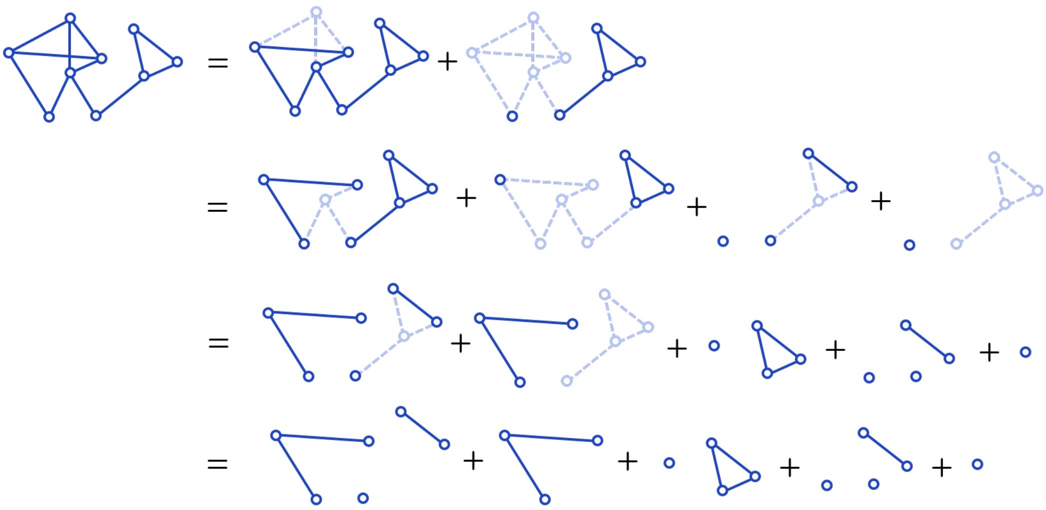



Figure 12: Independent set အရေအတွက် recurrence

ရလာတဲ့ network မှာရှိတဲ့ independent set အရေအတွက်ပါပဲ။ ဒီလိုပဲ $u$ ပါဝင်တဲ့ set အရေအတွက်က network ထဲကနေ $u$ နဲ့ $u$ ရဲ့ friend တွေကိုဖယ်ပြီးရလာတဲ့ network မှာရှိတဲ့ independent set အရေအတွက်ပါပဲ။ ဥပမာအနေနဲ့ figure 11 ရဲ့ ဘယ်ဘက်စွန်ဆုံးပုံမှာ independent set ဘယ်နှခုရှိသလဲဆိုတာကို figure 12 ထဲကအတိုင်းတွက်လို့ရပါတယ်။



$$= 5\cdot 3\cdot 2+5+2\cdot 4+2\cdot 2\cdot 3+2$$

Figure 13: Recurrence ကိုသုံးပြီး independent set အရေအတွက်ရှာပုံဥပမာ

**Example 3.9.** Let $k\geq 3$ be a positive integer. In how many ways can we paint the vertices of a regular pentagon so that adjacent vertices receive different colours?

**Solution.** Vertex တွေကို $A$, $B$, $C$, $D$, $E$ လို့နာမည်ပေးလိုက်။ တကယ်လို့များ edge $AB$ မရှိဘူးဆိုပါစို့။ ဆိုလိုတာက $A$ နဲ့ $B$ အရောင်တူခွင့်ရှိတယ်ဆိုပါစို့။ ဒါဆိုရင် မူရင်းပုံက vertex တွေကို အရောင်ခြယ်နိုင်တဲ့နည်းသည် edge $AB$ ကိုဖယ်ထားတဲ့ပုံမှာ vertex တွေကိုအရောင်ခြယ်နိုင်တဲ့နည်းအရေအတွက်ထဲက $A$ နဲ့ $B$ အရောင်တူတဲ့ခြယ်နည်းအရေအတွက်ကို နုတ်ထားတာပဲ။ ဒါပေမယ့် $A$ နဲ့ $B$ အရောင်တူတဲ့နည်းအရေအတွက်က vertex $A$ နဲ့ $B$ ကို ပူးပစ်ပြီး

ထွက်လာတဲ့ပုံ (quadrilateral $ABCD$) ရဲ့ vertex တွေကို အရောင်ခြယ်နိုင်တဲ့ နည်းအရေအတွက်နဲ့ အတူတူပါပဲ (see figure 14)။

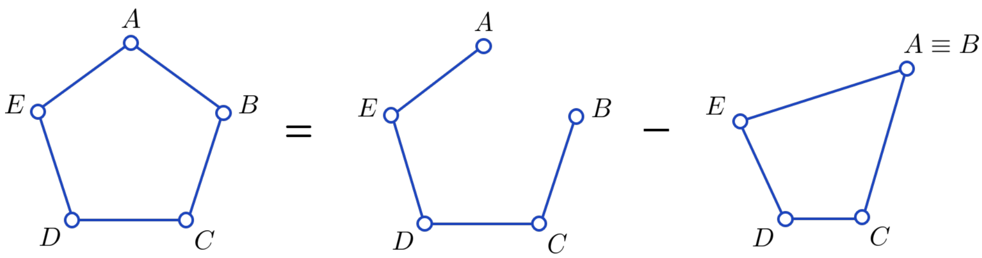



Figure 14: အရောင်ခြယ်ပုစ္ဆာအတွက် recurrence

Figure 14 ရဲ့ညာဘက်ခြမ်းထဲက ပထမပုံကို အရောင်ခြယ်နိုင်တဲ့နည်းသည် $k(k-1)^4$ ပဲ။ ဒုတိယပုံကိုခြယ်နိုင်တဲ့နည်းကိုတော့ စောနကလုပ်သလိုပဲ edge တစ်ကြောင်းဖြုတ်ပြီးဆက်တွက်သွားလို့ရပါပြီ (see figure 15)။

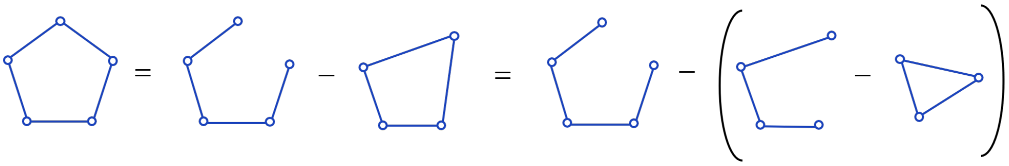


Figure 15: Recurrence ကို အဆင့်ဆင့်သုံးပြီးဖြေရှင်းမယ်

ဒါကြောင့်အဖြေက $k(k-1)^4 - (k(k-1)^3 - k(k-1)(k-2))$ ပဲဖြစ်ပါတယ်။ $\square$

---

**Quick Exercise 3.10.** Let $n, k \geq 3$ be positive integers. In how many ways can the vertices of a regular $n$-gon be coloured using a supply of $k$ colours so that adjacent vertices receive different colours?

ဒီပုစ္ဆာကိုလည်း example 3.8 လိုမျိုးပဲ network အကြီးတွေအတွက် အရောင်ခြယ်နည်းရှာတဲ့အခါ သုံးလို့ရပါတယ်။ Network တစ်ခုရဲ့ vertex (user) တွေကို friend အချင်းချင်းအရောင်မတူအောင် အရောင် $k$ ခုနဲ့ခြယ်ချင်တယ်ဆိုပါစို့ (အရောင်အကုန်လုံးသုံးစရာမလိုပါ)။ အရင်ဆုံး network ထဲကကြိုက်ရာ edge တစ်ခု $e$ ကိုကြည့်လိုက်ပါ။ ဒီ $e$ ကိုဖြုတ်ချလိုက်ရာကနေ ထွက်လာတဲ့ network ကိုအရောင်ခြယ်နိုင်သောနည်းက မူရင်း network ထက်ပိုများပါတယ်။ $e$ ကနေဆက်သွယ်ထားတဲ့ user နှစ်ယောက်က အရောင်တူလို့ရသွားတာကိုး။ ပိုလာတဲ့အရောင်ခြယ်နည်းအရေအတွက်က $e$ က ဆက်သွယ်ထားတဲ့ user နှစ်ယောက်ကို ပူးလိုက်ပြီးထွက်လာတဲ့ network ကိုအရောင်ခြယ်နိုင်တဲ့နည်းနဲ့ အတူတူပဲ။ ပုံနဲ့ပြရင် Figure 16 လိုမျိုးပေါ့။

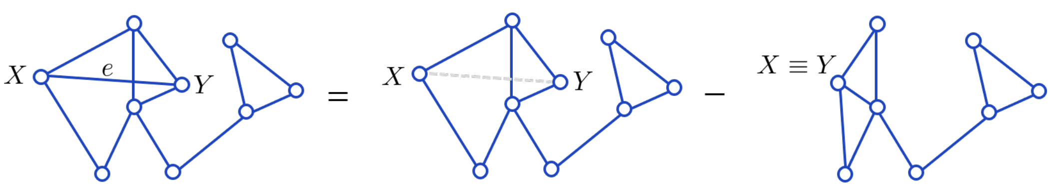



Figure 16: အရောင်ခြယ်နည်းအတွက် recursion

ဒီ recursion က တစ်ဆင့်လုပ်လိုက်တိုင်းလုပ်လိုက်တိုင်းမှာ edge အရေအတွက်ကတစ်ခုလျော့သွားပါတယ်။ ဒါ

ကြောင့် ဒီ recursion ကိုအဆင့်ဆင့်လုပ်ပြီးချိန်မှာ edge အရေအတွက်တွေက အရမ်းနည်းသွားမယ်။ Edge အရေအတွက် အတော်လေးနည်းသွားပြီဆိုရင်တော့ အရောင်ခြယ်နည်းကို လက်နဲ့ရှာလိုက်လို့ရပါပြီ။ ဘယ်လိုပုံတွေက လက်နဲ့ရှာလို့ ရသလဲဆိုရင် ဥပမာအနေနဲ့ Figure 11 ရဲ့ အလယ်ကပုံကို အရောင်ခြယ်နိုင်တဲ့ နည်းလမ်းအရေအတွက်ကိုရှာကြည့်ပါ။ အဖြေက $k(k-1)^{14}$ ပါ။ In general ပြောရင်တော့ cycle မပါတဲ့ network တွေဆိုရင်လွယ်လွယ်လေးရှာလို့ရပါတယ်။ ဒါကြောင့် network တစ်ခုကိုအရောင်ခြယ်နိုင်တဲ့နည်းကို တွက်ဖို့အတွက် cycle မပါတဲ့ပုံတွေချည်းကျန်တဲ့အထိ recursion ကို သုံး သုံး ပြီး လျှော့ သွား ရုံ ပါ ပဲ။ ဥပမာ အနေ နဲ့ Figure 11 ရဲ့ ဘယ် ဘက် စွန် ဆုံး ပုံ ကို အရောင် ခြယ် နိုင် တဲ့ နည်း လမ်း အရေအတွက်ကို Figure 17 အတိုင်းတွက်ယူလို့ရပါတယ်။

**Quick Exercise 3.11.** Draw your favourite map with not too many regions. In how many ways can you colour your map using $k$ colours (each colouring may not use all the colours) so that regions sharing a border receive different colours? Find the smallest possible value of $k$ for which the number of colourings is non-zero.

ဒီလောက်ဆိုရင် recursion တစ်ခုဖြစ်ဖို့အတွက် ကိန်းစဉ်ဖြစ်စရာမလိုဘူးဆိုတာကို ကွက်ကွက်ကွင်းကွင်းမြင်လောက်ပြီထင်ပါတယ်။ အရေးကြီးတာသည် ဒီ recursion ကိုထပ်ခါထပ်ခါသုံးသွားပြီးတဲ့အခါမှာ လက်နဲ့ရှာလိုက်လို့ရတဲ့ ပုစ္ဆာ အသေး လေး တွေ ဖြစ် သွား ဖို့ ပါ ပဲ။ ဒါ ကြောင့် recurrence relation တစ် ခု ဖြစ် ဖို့ အတွက် "တစ်ခုခု" ကို သေး သွား အောင် လုပ်နိုင်ရင်ရပါပြီ။ ဥပမာ independent set ရှာတဲ့ relation ဆိုရင် recurrence တစ်ခါသုံးပြီးတိုင်းသုံးပြီးတိုင်းမှာ vertex အရေအတွက်ကလျှော့သွားပါတယ်။ အရောင်ခြယ်တဲ့ recurrence မှာဆိုရင် edge အရေအတွက်ကလျှော့သွားပါတယ်။

## 4 Recursive Algorithms

Recursion ဆိုတာက အရေအတွက်ရေတဲ့နေရာမှာတင်သုံးတာမဟုတ်ပါဘူး။ ပုစ္ဆာကြီးတစ်ခုကို ပုံစံတူပုစ္ဆာငယ်လေး အဖြစ် ခွဲချလို့ရတိုင်းမှာ အသုံးပြုလို့ရပါတယ်။ ဘာကိုဆိုလိုသလဲသိရအောင် ပုစ္ဆာလေးနဲ့ ကြည့်ကြည့်ရအောင်။ အဖြေမကြည့်ခင်မှာ $n = 1, 2, 3, 4$ လောက်အတွက်ကို လက်နဲ့ချစမ်းကြည့်စေချင်ပါတယ်။

**Example 4.12.** In the game *Hanoi Tower*, you are given three pegs, and $n$ discs of increasing sizes are stacked through one of the pegs. Sizes of the discs decrease from top to bottom as in figure 18. In a *move*, you may take a disc from the top of a peg place it on a peg as long as you are not stacking a larger disc on top of a smaller one. You want to move all the discs to a peg that is different from the starting one. What is the minimum number of moves needed to do this?

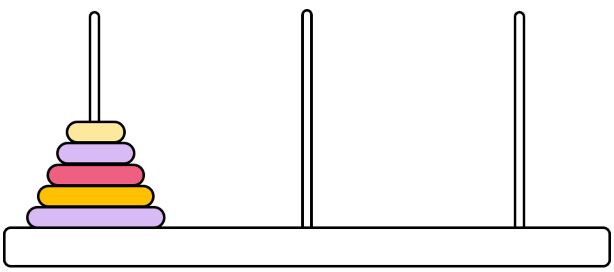


Figure 17: Hanoi Tower ဂိမ်း

**Solution.** အဖြေကို $a_n$ လို့ထားလိုက်ပါ။ အဝိုင်းပြား $n \geq 2$ ခုစလုံးကို တတိယတိုင်ကိုရွှေ့ချင်တယ်ဆိုပါစို့။ ဒါဆိုရင်

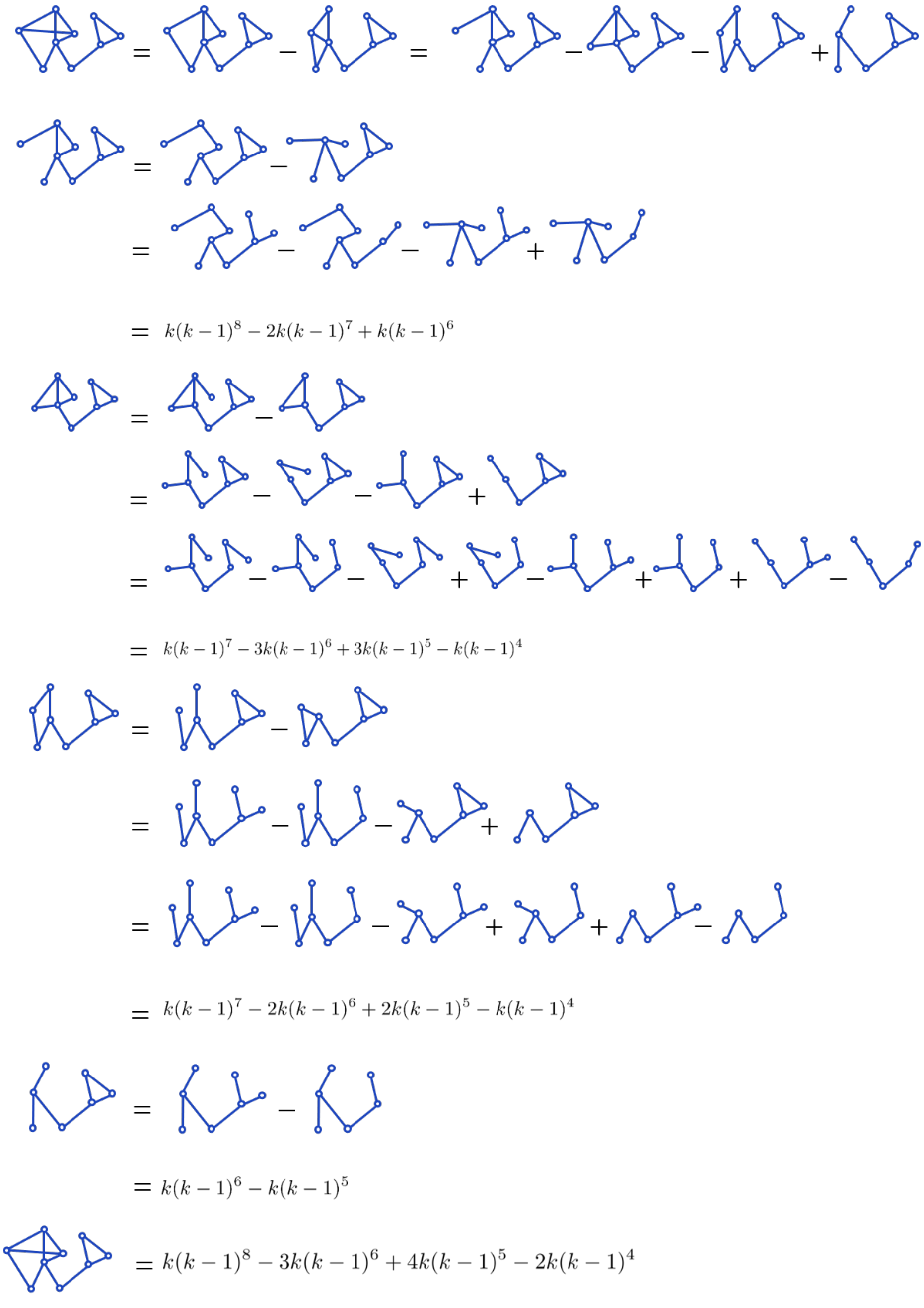



Figure 18: Recursion သုံးပြီးတွက်ပုံ ဥပမာ

အကြီးဆုံးအဝိုင်းပြားကို တတိယတိုင်ဆီ ရွှေ့နိုင်ဖို့လိုတယ်။ ဒါကိုလုပ်ဖို့အတွက် အကြီးဆုံးအဝိုင်းပြားပေါ်က အဝိုင်းပြား $n-1$ ခုကို ဒုတိယတိုင်ပေါ်ကို အရင်ရွှေ့ကိုရွှေ့ထားရလိမ့်မယ်။ ဒါကြောင့် အဝိုင်းပြား $n$ ခုစလုံးကိုတတိယတိုင်ကိုရွှေ့ဖို့အတွက်ဆိုရင် အောက်ပါအဆင့်သုံးဆင့်စလုံးကို မလုပ်မဖြစ်လုပ်ရလိမ့်မယ် (see figure 19)။

- အဆင့် (1)။ အငယ်ဆုံးအဝိုင်းပြား $n-1$ ခုကို ပထမတိုင်ကနေ ဒုတိယတိုင်ကိုရွှေ့ပါ။
- အဆင့် (2)။ အကြီးဆုံးအဝိုင်းပြားကို တတိယတိုင်ကိုရွှေ့ပါ။
- အဆင့် (3)။ အငယ်ဆုံးအဝိုင်းပြား $n-1$ ခုကို ဒုတိယတိုင်ကနေ တတိယတိုင်ပေါ်ကိုရွှေ့ပါ။

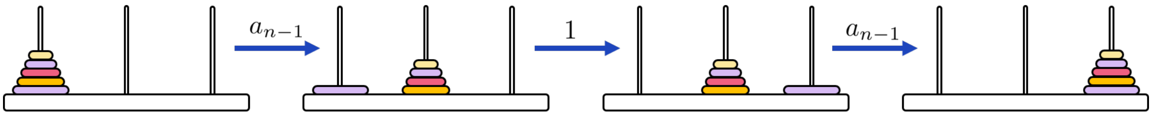



Figure 19: Hanoi Tower ကိုဖြေရှင်းဖို့ အဆင့်သုံးခု

အဆင့် (1) နဲ့ (3) ကိုလုပ်ဖို့ move အရေအတွက် $a_{n-1}$ စီလိုမယ်။ အဆင့် (2) က move တစ်ခုလိုတယ်။ ဒါကြောင့် $n \geq 2$ အားလုံးအတွက်

$$a_n = 2a_{n-1} + 1$$

ဖြစ်ရမှာပေါ့။ Initial condition က $a_1 = 1$ ဖြစ်တဲ့အတွက် $a_n$ ကိုရှာလို့ရပါပြီ။ ပြောင်းပြန် substitute ပြီးတွက်လိုက်လို့ရပါတယ် (ကိုယ်တိုင်စမ်းကြည့်ပါ)။ ဒါပေမယ့် ပိုသက်သာတဲ့နည်းကတော့ recurrence ရဲ့ နှစ်ဖက်စလုံးကို $1$ ပေါင်းပြီး

$$a_n + 1 = 2(a_{n-1} + 1)$$

ဖြစ်ကြောင်းမြင်ဖို့ပါပဲ။ ဒါကြောင့် $a_1 + 1, a_2 + 1, a_3 + 1, \ldots$ ဆိုတဲ့ကိန်းစဉ်က first term $2$ နဲ့ common ratio $2$ ရှိတဲ့ geometric progression ပဲ။ ဒါကြောင့် $a_n + 1 = 2^n$ ဖြစ်လို့ $a_n = 2^n - 1$ ပေါ့။ □

---

**Example 4.13.** There are sixteen sheep with distinct weights, and you have a sheep-scale. You can put two sheep on the sheep-scale at a time, and the sheep-scale tells you which sheep is heavier. We would like to arrange the sheep in increasing order by weights. Can we do this by using the sheep-scale less than 50 times?

**Solution.** သိုး $2^n$ ကောင်ကို ငယ်စဉ်ကြီးလိုက်စီဖို့အတွက် ချိန်ခွင်ကိုသုံးရမယ့် အကြိမ်အရေအတွက်ကို $a_n$ လို့ထားလိုက်မယ်။ ဒါဆိုရင် $a_1 = 1$ ပေါ့။ $a_n$ နဲ့ $a_{n-1}$ ကြားကဆက်သွယ်ချက်ကိုကြည့်ကြည့်ရအောင်။ သိုး $2^n$ ကောင်ရှိတယ်ဆိုပါစို့။ ဒီသိုးတွေကို $2^{n-1}$ ကောင်ပါအုပ်စုနှစ်ခု $A$ နဲ့ $B$ ဆိုပြီးခွဲလိုက်မယ်။ အုပ်စုတစ်ခုစီထဲကသိုးတွေကို ချိန်ခွင် $a_{n-1}$ ခါစီသုံးပြီးတော့ ငယ်စဉ်ကြီးလိုက်စီလိုက်လို့ရတယ်။ ဒါဆိုရင် ငယ်စဉ်ကြီးလိုက်စီထားတဲ့ အုပ်စု $A$ နဲ့ ငယ်စဉ်ကြီးလိုက်စီထားတဲ့ အုပ်စု $B$ နှစ်ခုကို ဘယ်လိုပေါင်းစပ်မလဲဆိုတာကိုပဲ ဆက်စဉ်းစားရတော့မယ်။ သိုးအားလုံးထဲက အပေါ့ဆုံးသိုးကို ချိန်ခွင်တစ်ခါပဲသုံးပြီးရှာလို့ရပါတယ် (ဘယ်လိုလဲ)။ အပေါ့ဆုံးသိုးကိုသိသွားရင် ချိန်ခွင်ကိုနောက်တစ်ခါထပ်သုံးပြီး ဒုတိယအပေါ့ဆုံးသိုးကို ဆက်ရှာလို့ရပါတယ် (ဘယ်လိုလဲ)။ ဒီတိုင်းပဲဆက်သွားရင် ချိန်ခွင်ကိုတစ်ခါစီသုံးပြီး လက်ကျန်သိုးတွေထဲကအပေါ့ဆုံးသိုးတွေကို ဆက်ဆက်ရှာသွားလို့ရပါတယ်။ ဒါကြောင့် ချိန်ခွင်ကို $2^n - 1$ ကြိမ်သုံးပြီးရင် သိုးအားလုံးကို

ငယ်စဉ်ကြီးလိုက်စီပြီးသားဖြစ်သွားမှာပေါ့။ ဒါကြောင့် $n \geq 2$ အားလုံးအတွက်

$$a_n \leq 2a_{n-1} + 2^n - 1$$

ဖြစ်တာကိုတွေ့ရပါမယ်။ ဒီနေရာမှာ $=$ မဟုတ်ဘဲ $\leq$ ဖြစ်နေတဲ့အကြောင်းရင်းကတော့ အခုတွက်လိုက်တာက ပြောနေတာသည် "ချိန်ခွင်ကို $2a_{n-1} + 2^n - 1$ ခါသုံးပြီး ငယ်စဉ်ကြီးလိုက်စီလို့**ရ**တယ်" ဆိုတာကိုပြောနေတာပါ။ အကြိမ်အရေအတွက် အငယ်ဆုံးဟုတ်မဟုတ်မသေချာပါဘူး။ ဒါပေမယ့် $a_n$ ဆိုတာကကျတော့ ချိန်ခွင်ကိုသုံးရမယ့် **အငယ်ဆုံး**အကြိမ်အရေအတွက်ပါ။ ဒါကြောင့် $\leq$ လို့ပဲပြောနိုင်တာဖြစ်ပါတယ်။ ဒါပေမယ့်ဒီလောက်ရပြီဆိုရင် ပုစ္ဆာကမေးတဲ့မေးခွန်းကိုဖြေလို့ရပါပြီ။

$$a_4 \leq 2a_3 + 15 \leq 2(a_2 + 7) + 15 \leq 2(2(a_1 + 3) + 7) + 15 = 49$$

ဒါကြောင့် ချိန်ခွင်ကို $49$ ခါပဲသုံးပြီး သိုး $16$ ကောင်ကိုငယ်စဉ်ကြီးလိုက်စီလို့ရပါတယ်။ □

---

**Remark.** "နှိုင်းယှဉ်ခြင်းကိုပဲသုံးပြီးတော့ ပေးထားတဲ့ကိန်းတွေကို ငယ်စဉ်ကြီးလိုက်ဘယ်လိုစီမလဲ" ဆိုတာက computer science မှာသိပ်ကိုအရေးကြီးတဲ့ မေးခွန်းတစ်ခုပါ။ ဒီပုစ္ဆာမှာသုံးသွားတဲ့ ငယ်စဉ်ကြီးလိုက်စီနည်းကို merge sort လို့ခေါ်ပါတယ်။ In general ပြောရရင် သိုးအရေအတွက် $N$ ကောင်ရှိရင် ဒီနည်းကိုသုံးပြီးတော့ ချိန်ခွင်ကို $2N\log_2(N) + 1$ ထက်ပိုမသုံးဘဲ ငယ်စဉ်ကြီးလိုက်စီလို့ရပါတယ်။

ဒီ section ကပုစ္ဆာတွေက အရှေ့ section နှစ်ခုနဲ့ယှဉ်ရင် သိသိသာသာပိုခက်တာကို တွေ့ရမှာပါ။ အကြောင်းရင်းကတော့ ရှေ့ section တွေကပုစ္ဆာတွေက ပေးထားတဲ့အရာဝတ္ထုတစ်ခုကို လေ့လာရတဲ့ပုစ္ဆာမျိုးဖြစ်ပြီး ဒီ section ကပုစ္ဆာတွေကကျတော့ လိုချင်တာတစ်ခုကို ဘယ်လိုတည်ဆောက်ရမလဲဆိုတာ ကိုယ်တိုင်စဉ်းစားရလို့ဖြစ်မယ်လို့ ထင်ပါတယ်။ လုပ်လို့ရတာတွေပိုများနေတော့ ပုစ္ဆာကလည်းပိုခက်သွားတာပေါ့။ ဘယ်လိုပဲဖြစ်ဖြစ် recursion ဆိုတဲ့အိုင်ဒီယာတစ်ခုလုံးက ပုစ္ဆာအကြီးကြီးကို ပုံစံတူပုစ္ဆာငယ်လေးတွေဖြစ်အောင် ပြောင်းဖို့ကြိုးစားတာပါပဲ။

## 5 Exercises

ဒီ exercise အားလုံးမှာ ဘာမှထူးထွေပြောမထားဘူးဆိုရင် $n$ သည် အမြဲတစေ positive integer တစ်ခုဖြစ်ပါတယ်။ $*$ နဲ့ပြထားတဲ့ပုစ္ဆာတွေက ပိုခက်ပါတယ်။

1. Find the formula for $a_n$ in each of the following recurrence relations:

   (a) $a_1 = 1$, $a_n = a_{n-1} - 3$ for all $n \geq 2$,

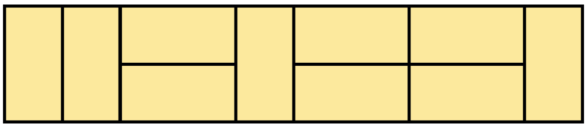
   (b) $a_1 = 3$, $a_n = \frac{1}{2}a_{n-1}$ for all $n \geq 2$,

   (c) $a_1 = 2$, $a_n = a_{n-1} + 2n - 1$ for all $n \geq 2$,

   (d) $a_1 = 0$, $a_n = 3a_{n-1} + 1$ for all $n \geq 2$,

   (e) $a_1 = 0$, $a_2 = 1$, $a_n = 2a_{n-1} - a_{n-2} + 1$ for all $n \geq 3$.

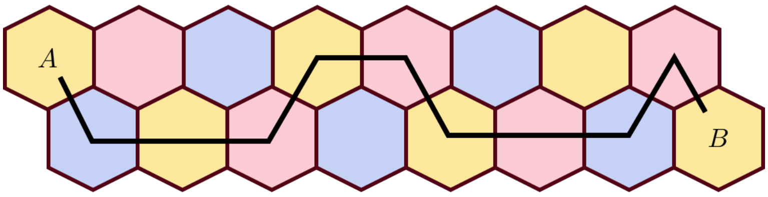
2. Find the number of ways to tile a $2 \times 10$ rectangle using identical $1 \times 2$ and identical $2 \times 1$ rectangles.



3. Jumpy wants to walk up a flight of $6$ stairs. In each step, he can climb either $1$, $2$ or $3$ stairs. In how many different ways can he climb the flight of stairs?

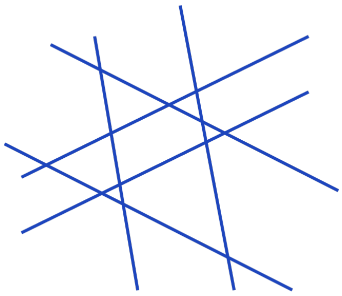
4. Norman wishes to buy a can of soda costing $75$ cents from a vending machine. He has an unlimited supply of identical nikels (worth $5$ cents each) and dimes (worth $10$ cents each). In how many different orders can he insert coins into the machine to pay for his soda?

5. How many paths are there from the hexagon $A$ to hexagon $B$ in the diagram below, if each step of a path must be to a hexagon immediately adjacent on the right?




6. Paula works for a valet parking company. Each of our customers drives either a Cadillac, a Continental, or a Porsche. Paula's boss told her that she have to reserve spaces in the parking lot and mark them as being for a Cadillac, a Continential, or a Porsche. Cadillacs and Continentials each take $2$ spaces while Porsches only require $1$. If the parking lot has $12$ spaces, in how many ways can Paula allocate the spaces?

7. If we draw $n$ pairs of parallel lines in the plane, what is the largest number of different regions can we create?



8. Call a set of integers spacy if it contains no more than one out of any three consecutive integers. How many subsets of $\{1, 2, 3, \ldots, 12\}$ are spacy?

9. Let $a_n$ be the number of $n$-digit numbers that uses the digits $1$, $2$ and $3$ only so that $11$ does not appear at any part of the number. Find the initial conditions and the recurrence relation for $a_n$.

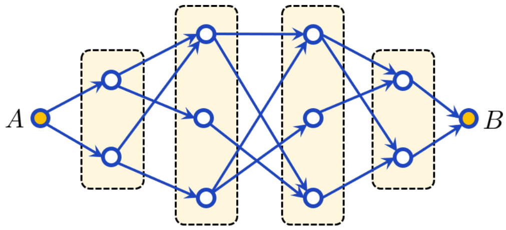
10. A $2 \times n$ rectangle is to be paved with $1 \times 2$ identical blocks and $2 \times 2$ identical blocks. Let $a_n$ denote the number of pavings. Find the initial conditions and the recurrence relation for $a_n$.

$*$ 11. Throughout the first semester, each of the $n$ students are assigned a unique book to read. When the second semester starts, these books are shuffled among the students so that no one gets the book they read in the first semester. Let $D_n$ be the number of such shufflings. Prove that

$$D_n = (n-1)(D_{n-1} + D_{n-2}).$$

$*$ 12. What is the sum of the greatest odd divisors of the integers $1, 2, 3, \ldots, 2^n$?

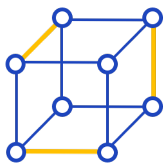
13. In the figure, find the number of paths from vertex $A$ to $B$ that follows the arrows.



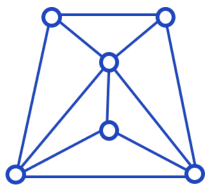

14. Find the number of ways to paint $4$ vertices of a regular $10$-gon so that no two coloured vertices are adjacent.

15. In how many ways can each cell of a $3 \times 3$ grid be coloured with red, blue or yellow so that no two adjacent cells have the same colour? Here, two cells are considered adjacent if they share an edge.

16. In a graph (network), a matching is a (possibly empty) set of edges no two of which shares a vertex. The figure shows an example of a matching (orange edges). How many matchings are there in the graph shown in the figure?



17. In a social network, a clique is a (possibly empty) set of users any two of which are friends with each other. Find the number of cliques in the following social network where the vertices represent users and edges represent friendships between the users.

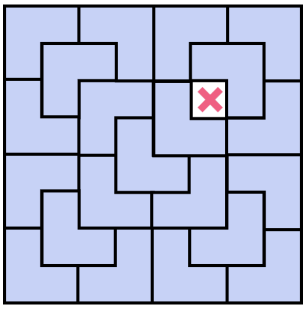


$*$ 18. Suppose we are given $1000$ lamps and $1000$ switches with each connected to one lamp and each lamp to one switch, but we do not know which lamp corresponds to which switch. Initially, all lamps are off. One operation consists of the following:

- We first specify an arbitrary set of switches, and all of them will be switched from off to on simultaneously.
- We will then see which lamps come on, then switch all the lamps off.

Show that we can determine which switch corresponds to which lamp within $10$ operations.

$*$ 19. Consider a $2^n \times 2^n$ chessboard. Show that this chessboard can be fully tiled with the L-shaped tiles as shown below (for $n = 3$).



$**$ 20. Prove that it is possible to place $3 \cdot 2^n$ points on the plane so that at least $3 \cdot 2^{n-1} \cdot (n+2)$ unordered pairs among them are distance $1$ apart.

$**$ 21. Let $S$ be a set with $2024$ elements, and let $N$ be an integer with $0 \leq N \leq 2^{2024}$. Prove that it is possible to colour every subset of $S$ either black or white so that the following conditions hold:

- the union of any two white subsets is white,
- the union of any two black subsets is black,
- there are exactly $N$ white subsets.

$**$ 22. Show that there exists a subset $A$ of $\{1, 2, 3, \ldots, 3^n\}$ of size $2^n$ such that $A$ does not contain a non-trivial arithmetic progression (that is, if $x, y, z \in A$ are distinct, then $z - y \neq y - x$).