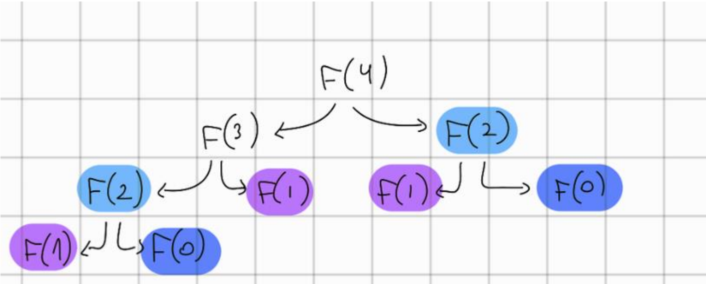
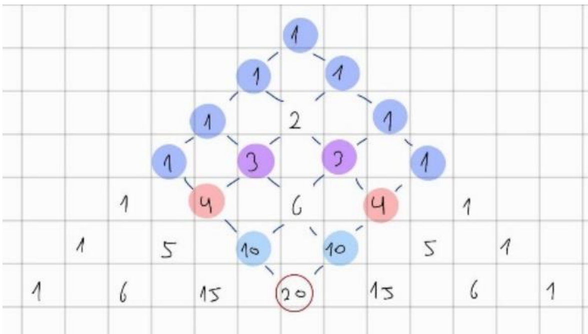
תכנות דינאמי

יש פעמים שהפתרון הרקורסיבי הוא פשוט אבל לא יעיל.
לדוגמה מספרי פיבונאצ'י: הזמן לחישוב $F(n)$ הוא בעצמו $F(n)$, שזה בערך $\Phi^n = O(2^n)$. נשים לב שבתהליך החישוב, אנחנו חוזרים על אותם חישובים:





גם בחישוב של $\binom{n}{k}$, כדי לא לחשב עצרת אפשר לחשב בנוסחה הרקורסיבית:
$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. אבל גם בשיטה הזאת, יוצא שחוזרים על אותם חישובים. זמן ריצה $O(\frac{2^n}{\sqrt{n}})$

אם יש פתרון רקורסיבי שמבצע את אותו חישוב הרבה פעמים, אפשר לייעל את זמן הריצה עם תכנות דינאמי – מתחילים ממקרה הבסיס ועולים למעלה לפי הנוסחה הרקורסיבית. במספרי פיבונאצ'י אפשר להתחיל מ $F(0)$ ולעלות. $O(n)$.
בחישוב $\binom{n}{k}$ אפשר למלא טבלה שבה $\binom{n}{k}$ זה העמודה ה k בשורה ה n. $O(n^2)$

באופן כללי, השיטה לתכנות דינאמי:
- נאפיין את המבנה של הפתרון האופטימלי והנוסחה הרקורסיבית.
- נחשב את הפתרון האופטימלי מלמטה-למעלה.
- נחזור אחורה לפי הנוסחה הרקורסיבית כדי לשחזר את פרטי הפתרון האופטימלי.

<u>טורניר</u>: שתי קבוצות A, B. בכל משחק יש לA הסתברות q לנצח (באופן בת"ל). הקבוצה הראשונה שמנצחת n משחקים מנצחת בטורניר. נגדיר $\mathbb{P}(i,j)$ ההסתברות ש A תנצח בטורניר, כאשר A צריכה עוד i נצחונות ו-B צריכה j. הנוסחה הרקורסיבית היא:
$$\mathbb{P}(0,j) = 1, \quad \mathbb{P}(i,0) = 0\ (i > 0),$$
$$\mathbb{P}(i,j) = q \cdot \mathbb{P}(i-1,j) + (1-q) \cdot \mathbb{P}(i,j-1)$$
אפשר לחשב עם טבלה: נתחיל ככה ונמלא משמאל לימין לפי שורות. $O(n^2)$.

i \ j	0	1	2	⋯
0	∅	1	1	⋯
1	0			
2	0			
⋮	⋮			

<u>כפל מטריצות</u>: עבור מטריצות: $\underbrace{A}_{p\times q} \cdot \underbrace{B}_{q\times r} = \underbrace{C}_{p\times r}$. לכל תא בC נכפיל עמודה בשורה, כל אחת q תאים. אז סה"כ $O(p \cdot q \cdot r)$ פעולות כפל. מכיוון שכפל מטריצות הוא אסוציאטיבי, בהינתן 3 מטריצות נוכל לבחור מה להכפיל קודם ולהקטין את מספר הפעולות הנדרשות. לדוגמה:

$$\underbrace{\underbrace{A}_{2\times5} \cdot \underbrace{\left(\underbrace{B}_{5\times3} \cdot \underbrace{C}_{3\times4}\right)}_{60}}_{} = \underbrace{\underbrace{A}_{2\times5} \cdot \underbrace{BC}_{5\times4}}_{40} = \underbrace{ABC}_{2\times4}$$ סה"כ 100

$$\underbrace{\left(\underbrace{A}_{2\times5} \cdot \underbrace{B}_{5\times3}\right)}_{30} \cdot \underbrace{C}_{3\times4} = \underbrace{\underbrace{AB}_{2\times3} \cdot \underbrace{C}_{3\times4}}_{24} = \underbrace{ABC}_{2\times4}$$ סה"כ 54

עבור מטריצות $M_1 \cdots M_n$ כך ש: $\underbrace{M_i}_{d_{i-1}\times d_i}$, נגדיר $m(i,j)$ מספר הפעולות המינימלי להכפלת $M_i \cdots M_j$. הנוסחה הרקורסיבית:
$$m(i,j) = \begin{cases} 0, & i = j \\ \min\limits_{i\le k\le j} \{\underbrace{m(i,k)}_{left} + \underbrace{m(k,j)}_{right} + \underbrace{d_{i-1} \cdot d_k \cdot d_j}_{p\cdot q\cdot r}\} \end{cases}$$
נוכל לחשב את זה בתכנות דינאמי, עם 2 טבלאות:

s	1	2	3
1		1	2
2			1
3			

m	1	2	3
1	0	30	54
2		0	60
3			0

S זה נקודת הפיצול האופטימלית.
באופן כללי, צריך למלא $n^2/2$ תאים, ובכל אחד למצוא את המינימום מבין $O(n)$ אופציות. סה"כ $O(n^3)$.

<u>בעיית תרמיל הגב בשלמים</u>: כמו הבעיה בשברים, רק שאי אפשר לקחת חלק מפריט.

$\sum_{i=1}^{n} f_i v_i,\ \ \text{such that:} f_i \in \{0,1\},\ \sum_{i=1}^{n} w_i f_i \le C$. הנוסחה הרקורסיבית היא: הערך המקסימלי עבור i פריטים במשקל B:

$$T(i,B) = \max\{\underbrace{T(i-1,B)}_{\text{if we don't take } G_i},\ \underbrace{T(i-1,B-w_i)+v_i}_{\text{if we take } G_i}$$

דוגמה:

i	0	1	2	3	4	5
v_i	4	8	7	1	2	10
w_i	9	6	3	4	5	2

weight limit / up to item	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	4	4	4	4	4	4	4
2	0	0	0	0	0	0	8	8	8	8	8	8	8	8	8	12
3	0	0	0	7	7	7	8	8	8	15	15	15	15	15	15	15
4	0	0	0	7	7	7	8	8	8	15	15	15	15	16	16	16
5	0	0	0	7	7	7	8	8	9	15	15	15	15	16	17	17
6	0	0	10	10	10	17	17	17	18	18	19	25	25	25	25	26

נמלא את השורה ועמודה הראשונות, ואז נמלא את הטבלה לפי השורות משמאל לימין.
כדי לשחזר את הפתרון עצמו (איזה פריטים לקחת) נחזור אחורה לפי הנוסחה. אם התא מעלינו שווה לנו, זה אומר שלא לקחנו את הפריט הזה. ואם לא, כלומר הגענו מהתא השמאלי יותר, זה אומר שכן לקחנו את הפריט.

זמן ריצה: $O(C \cdot n)$. כלומר תלוי **בערך** של הקלט, ולא רק האורך. זה נקרא זמן ריצה **פסודופולינומי.**

<u>תת-מערך מקסימלי</u>: בהפרד ומשול היה פתרון $O(n\log n)$.
בתכנות דינאמי: לכל $1 \le i \le n$ נמצא את תת-המערך המקסימלי עד המקום ה-i.
$T[i] = \max\{\underbrace{T[i-1]+A[i]}_{\text{add } A[i] \text{ to sub-array}},\ \underbrace{A[i]}_{\text{take only } A[i]},\ \underbrace{0}_{\text{take nothing}}\}$. נשמור מערך T ונרוץ על A. $O(n)$.

<u>תת-מטריצה מקסימלית</u>: פתרון נאיבי – לרוץ על כל האופציות, $O(n^6)$
פתרון ב – מערך עזר: לכל תת-רצף של שורות (יש $\binom{n}{2}$ כאלה), נייצר מערך שמחזיק את הסכום של תת-המטריצה הזאת. בעזרת הפונקציה של תת-מערך מקסימלי נמצא את תת-המערך המקסימלי מביניהם. סה"כ $O(n^4)$

2	1	-3	-4	5
0	6	3	4	1
2	-2	-1	4	-5
-3	3	1	0	3

2	4	2	8	-4

פתרון ג – מטריצת עזר: נבנה מטריצה H שתחזיק בתא $H[i][j]$ את הסכום של A עד התא הזה. בעזרת הכלה והדחה, נוכל לחשב את זה ביעילות יחסית. בשורה ועמודה הראשונות:

$$H[0][0] = A[0][0], \qquad H[0][j] = H[0][j-1] + A[0][j], \qquad H[i][0] = H[i-1][0] + A[i][0]$$

ובכל שאר התאים: $H[i][j] = \underbrace{A[i][j]}_{3} + \underbrace{H[i][j-1]}_{7} + \underbrace{H[i-1][j]}_{-2} - \underbrace{H[i-1][j-1]}_{1}$.

A:

1	-3
6	3

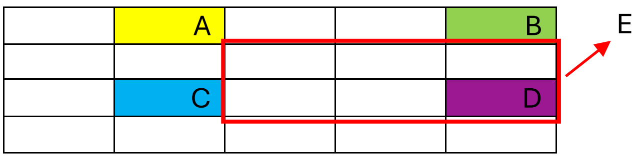
H:

1	-2
7	7

כדי לחשב תת מטריצה, נעשה הכלה והדחה על H:

$$E = D - C - B + A$$

זה גם $O(n^4)$



<u>סכום תת-קבוצה</u>:
בהינתן קבוצה A, האם קיימת תת קבוצה בסכום M?
פתרון רקורסיבי - $O(2^n)$. תכנון דינאמי – טבלה: $O(n \cdot m)$.

```
for(i=0 → n){
    for(j=0 →m){
        if (A[i] > j) H[i][j] ←H[i-1][j]
        else H[i][j] ← H[i-1][j] || H[i-1][j-A[i]]
```

A= {2, 5, 3, 4} , m= 6

H=

i \ j	0	1	2	3	4	5	6
φ	T	F	F	F	F	F	F
{2}	T	F	T	F	F	F	F
{2,5}	T	F	T	F	F	T	F
{2,5,3}	T	F	T	T	F	T	F
{2,5,3,4}	T	F	T	T	T	T	T

<u>מרחק עריכה – edit distance</u>:
עבור 2 מחרוזות, מספר הפעולות הדרוש כדי להפוך אחת לשנייה. פעולות מותרות: 1) שינוי תו, 2) מחיקת תו, 3) הוספת תו.

פתרון רקורסיבי: אם התווים במיקום האחרון שווים, נחזיר: קריאה רקורסיבית על שני המילים בלי התו האחרון.
אם המילים שונות בתו האחרון: נחזיר 1+ המינימלי מבין: הוספת תו, מחיקת תו, שינוי תו.

	ϕ	S	su	sun	sund	sunda	sunday
ϕ	0	1	2	3	4	5	6
s	1	0	1	2	3	4	5
Sa	2	1	1	2	3	3	4
sat	3	2	2	2	3	4	4
satu	4	3	2	3	3	4	5
satur	5	4	3	3	4	4	5
saturd	6	5	4	4	3	4	5
saturda	7	6	5	5	4	3	4
saturday	8	7	6	6	5	4	3

$\mathrm{ED}(A, B, m, n)$

$\quad \mathrm{if}(A[m] == B[n]) \text{ return } \mathrm{ED}(A, B, m-1, n-1);$

$\quad \text{else return } 1 + \min\{\underbrace{\mathrm{ED}(A, B, m, n-1)}_{\text{insert}}, \underbrace{\mathrm{ED}(A, B, m-1, n)}_{\text{delete}}, \underbrace{\mathrm{ED}(A, B, m-1, n-1)}_{\text{replace}}\}$