<u>תכנות דינאמי</u>
דוגמה ראשונה: אלגוריתם פלויד-וורשל למציאת כל המסלולים הקלים ביותר בגרף.

דוגמה שניה: מציאת קבוצה בלתי תלויה מקסימאלית עם משקלות (קודם כל משקל מקסימאלי ואם יש שני פתרונות עם אותו המשקל, אז את זה שיש בו הכי הרבה קטעים).
לדוגמה, קבוצת תורמים מגיעה ליום הרצאות באולם הרצאות, וידוע עבור כל תורם כמה כסף הוא יתרום אם הוא יקבל הרצאה. בכל רגע יכול להרצות תורם אחד בלבד, ולכל תורם ידוע זמן ההתחלה של ההרצאה שלו וזמן הסיום שלה. התורם יתרום רק אם הוא יקבל הרצאה. המטרה היא לבחור בתת קבוצה של התורמים שתכניס הכי הרבה כסף, כך שאין זמן שבו שנים או יותר מהתורמים שבקבוצה מרצים בו.

<u>קלט:</u> אוסף של מטלות: $s=\{a_1,a_2,\dots,a_n\}$, כל מטלה מאופיינת ע"י קטע (s_i,f_i) (זמן התחלה וזמן סיום), ומשקל w_i. הקטעים ממוינים בסדר עולה לפי זמן הסיום f_i.
<u>פלט:</u> תת קבוצה A של המטלות, שאף שתים מהן לא נחתכות, ושסכום משקליהן מקסימאלי.

<u>נגדיר:</u> $opt(i)=$ משקל אופטימאלי עבור המשימות $a_1,\dots,a_i$.
$opt(0)=0$
$[A(i)=$ קבוצת מטלות שנותנות את המשקל $opt(i)]$
המשקל $opt(i)$ - או שהוא לא משתמש ב a_i, ואז הוא שווה ל $opt(i-1)$,
או שהוא כן משתמש ב a_i ואז $opt(i)$ משתמש רק ב a_i ובקטעים שמסתיימים לפני s_i.
<u>הגדרה:</u> $pred(i)=\begin{cases}\max\{j \mid f_i\le s_i\}\\ 0\end{cases}$ (אם קיים j כזה אז המקסימאלי, או 0 אם לא קיים).
<u>טענה:</u> לכל $i>0$: $opt(i)=\begin{cases}opt(i-1)\\ opt(pred(i))+w_i\end{cases}$ (אחת משתי האפשרויות).
<u>הוכחה:</u> אם קיים פתרון אופטימאלי שלא משתמש ב a_i, אז $opt(i)=opt(i-1)$, שכן זהו גם פתרון אופטימאלי עבור $i-1$.
אחרת, $opt(i)$ משתמש ב a_i, ושאר הקטעים התורמים לו מהווים פתרון אופטימאלי ל $opt(pred(i))$.

<u>אלגוריתם תכנות דינאמי למציאת קבוצה בלתי תלויה בעלת משקל מקסימאלי:</u>
אתחול: מיין את המטלות לפי סדר עולה של זמני סיום: $f_1\le f_2\le\dots\le f_n$. $O(n\log n)$
לכל i, מצא את $pred(i)$, $A(0)\leftarrow\phi$, $opt(0)\leftarrow 0$. $O(n\log n)$ (חיפוש בינארי n פעמים).
$opt(i),A(i)$ - פתרון אופטימאלי עבור $a_1,\dots,a_i$.
עבור $i=1$ עד $i=n$ בצע: $O(n)$
אם $opt(pred(i))+w_i>opt(i-1)$ אז $O(1)$
$A(i)\leftarrow A(pred(i))\cup\{a_i\}$ $O(1)$
$opt(i)\leftarrow opt(pred(i))+w_i$ $O(1)$
אחרת
$A(i)\leftarrow A(i-1)$ $O(1)$
$opt(i)\leftarrow opt(i-1)$ $O(1)$
סה"כ סיבוכיות: $O(n\log n)$

דוגמה שלישית: סדר אופטימאלי למכפלת n מטריצות.

<u>מכפלת שתי מטריצות:</u>

$A_{m\times n}\times B_{n\times r}=C_{m\times r}$

$$\begin{pmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{pmatrix} X \begin{pmatrix} y & y & y & y \\ y & y & y & y \\ y & y & y & y \end{pmatrix} = \begin{pmatrix} z & z & z & z \\ z & z & z & z \\ z & z & z & z \\ z & z & z & z \\ z & z & z & z \end{pmatrix}$$

<u>מכפלת שלוש מטריצות:</u>

$A_{10\times 100}B_{100\times 5}C_{5\times 50}$

ניתן לבצע בשתי צורות:

1. $\left[\left(A_{10\times 100}B_{100\times 5}\right)C_{5\times 50}\right]$ - 5,000 פעולות עבור $A\times B$ ועוד 2,500 עבור המכפלה ב C.
סה"כ 7,500 פעולות.
2. $\left[A_{10\times 100}\left(B_{100\times 5}C_{5\times 50}\right)\right]$ - 25,000 פעולות עבור $B\times C$ ועוד 50,000 עבור המכפלה ב A.
סה"כ 75,000 פעולות.

<u>קלט:</u> סדרה של n מטריצות $\left(A_1,\ldots,A_n\right)$ מטריצה A_i ממימדים $p_{i-1}\times p_i$

<u>פלט:</u> מספר מכפלות בעץ אופטימאלי (או סידור סוגריים אופטימאלי) למכפלת n המטריצות.
(בפועל האלגוריתם ניתן "לשיפוץ" כך שיחזיר גם את הסדר האופטימאלי הנחוץ).

עבור: $1\le i<j\le n$ נגדיר $m\left[i,j\right]$ - מספר המכפלות המינימאלי לביצוע הכפל: $A_i\times\ldots\times A_j$.

מחפשים את $m\left[1,n\right]$.

$$m\left[1,n\right]=\min\left\{m\left[1,k\right]+m\left[k+1,n\right]+p_0\cdot p_k\cdot p_n \mid 1\le k\le n\right\}$$

סיבוכיות של מימוש רקורסיבי נאיווי:

$$T\left(n\right)\ge 2\sum_{i=1}^{n-1}T\left(i\right)+2\left(n-1\right)>2^{n-1}$$

פתרון יותר יעיל:

תכנות דינאמי לכפל אופטימאלי של n מטריצות:

אתחול: עבור $i=1$ עד $i=n$ בצע: $m[i,i]=0$ $O\left(n\right)$

עבור $j=1$ עד $j=i-1$ בצע: ($j+1=$ אורך תת השרשרת של המטריצות). $O\left(n\right)$

עבור $l=1$ עד $l=n-j$ בצע: ($l=$ האינדקס של המטריצה). $O\left(n\right)$

$O\left(n\right)$ $m[l,l+j]=\min\left\{m\left[l,k\right]+m\left[k,l+j\right]+p_{l-1}p_k p_{l+j} \mid l\le k<l+j\right\}$

סה"כ סיבוכיות: $O\left(n^3\right)$