**תכנון דינמי**

* תכנון דינמי פותר בעיות אופטימיזציה על ידי שימוש בצירופי פתרונות של תתי בעיות.
* המילה "תכנון" מתייחסת לבניית טבלה ובה פתרונות של תתי בעיות, אין הכוונה לקוד.
* גישת התכנון הדינמי כוללת 4 שלבים בפתרון הבעיה:
  + נאפיין את מבנה הפתרון האופטימלי.
  + נגדיר רקורסיבית את הערך של הפתרון האופטימלי.
  + נחשב את הערך של פתרון אופטימלי מלמטה למעלה.
  + נבנה פתרון אופטימלי מהמידע המחושב –   
    פתרון של תתי הבעיות הרלוונטיים.

**בעיית תכנון הרכבת מכונית:**

* נתון מפעל ובו שני פסי יצור בשיטת הסרט הנע.
* בכל פס יצור יש n תחנות יצור.
* שלד המכונית מגיע לתחנה הראשונה, והמכונית יוצאת מהפס אחרי התחנה ה- n.
* בכל פס יצור נעשית עבודה זהה, אך זמן העבודה שונה לכל תחנה בפסים השונים.
* בנוסף, העברת מכונית מפס יצור אחד לשני דורשת זמן נוסף.
* המטרה: מציאת אופן תכנון הרכבת המכונית במינימום זמן.



**קלט-הנתונים:**

* Sij: תחנה j בפס יצור i (j ∈ {1..n} , i ∈ {1,2}).
* הזמן לביצוע עבודה בתחנה Sij מסומן ע"י aij.
* הזמן להעברת מכונית מתחנה Sij לתחנה הבאה בפס השני: tij.
* זמן הכנסת שלד המכונית לתחנה הראשונה בפס i הוא ei.
* זמן הוצאת מכונית בנויה מהפס ה-i הוא xi.
* יש למצוא את המסלול המהיר ביותר להרכבת המכונית.

**פתרון נאיבי:**

לבדוק את כל אפשרויות המעבר בין שני הסרטים. יש בסך הכול n2 אפשרויות כאלה ולכן זאת שיטה לא סבירה.

**פתרון:**

נסמן ב- f[i,j] את הזמן המינימלי הנדרש כדי להגיע לתחנה Si,j ו- pred[i,j] ∈ {1,2} את התחנה הקודמת הנדרשת כדי להגיע לתחנה Si,j בזמן המינימלי.

נגדיר את הפונקציה f באופן רקורסיבי.

f[1,j] = { if j == 1: e1 +a1,1 , else: min( f[1,j-1], f[2,j-1] +t2,j-1) + a1,j }

f[2,j] = { if j == 1: e2 +a2,1, else: min( f[2,j-1] , f[1,j-1] +t1,j-1) + a2,j }

pred[1,j] = { if f[1,j-1] ≤ f[2,j-1] + t2,j-1 : 1, else: 2}

pred[2,j] = { if f[2,j-1] ≤ f[1,j-1] +t1,j-1 : 2, else : 1}

last = { if f[1,n] + x1 < f[2,n] + x2 : 1, else : 2}

אם נריץ את האלגוריתם הנ"ל בצורה רקורסיבית (מהסוף להתחלה) זמן הריצה שיתקבל הוא θ(2n) .

לכך נבנה את הפתרון מההתחלה לסוף ע"י שימוש בשני מטריצות f, pred בגודל 2 X n . באופן הבא: נאתחל את f[1][1] = a1,1 , f[2][1] = a2,1 . נעבור בלולאה מ- j=2 ועד n ונחשב את f[1][j] , f[2][j] לפי הנוסחה לעיל וכיוון שהערכים כבר חושבו לא נצטרך לחשב אותם שוב, בכל שלב נחשב גם את ה- pred[i][j] שיקבע לפי הנוסחה לעיל . לאחר שעברנו בלולאה וקבענו את ערכו של ה- pred נחשב את ערכו של last . אם last = i נדפיס i ונעבור pred[i][n] וכן הלאה מ- n ועד 1 . הקלט הרצוי הוא מה שהדפסנו אבל הפוך.