בס"ד

**תכנות דינאמי**

תכנות דינאמי נועד כדי לפתור בעיות שהפתרון הרקורסיבי שלהן הוא אקספוננציאלי. אך מצד שני מספר תתי הבעיות שחוזרות על עצמן הוא קטן.

הרעיון הוא לשמור את הפתרונות בטבלה בסדר כזה שלא נצטרך את הקריאות הרקורסיביות. כלומר, כאשר נגיע לתא מסוים בטבלה הפתרונות שלו יהיו בטבלה במקומות שכבר היינו בהם.

דוגמה 1: פיבונאצי

הפונקציה הרקורסיבית:

fib(n)

if(n=1 or n=2)

return 1

else

return fib(n-1)+fib(n-2)

fib(10)

fib(8)

fib(9)

fib(6)

fib(7)

fib(6)

fib(7)

fib(7)

fib(8)

אלגוריתים תכנון דינאמי:

1. מה יהיה גודל הטבלה – גדול הטבלה תלוי במספר הפרמטרים ש**משתנים** במהלך הקריאות הרקורסיביות (אם פרמטר אחד משתנה נגדיר מערך חד מימדי, אם שני פרמטרים משתנים נגדיר מערך דו מימדי, אם שלושה פרמטרים משתנים נגדיר מערך תלת מימדי וכו').

במקרה שלנו, הפרמטר המשתנה הוא n לכן יש לבנות טבלה חד מימדית בגודל n: A[n].

1. באיזה סדר לסרוק את הטבלה.
2. מה למלא בטבלה:
3. יש לקבוע מה הפרמטרים שירוצו על הטבלה: לכל פרמטר ש**משתנה** ברקורסיה נגדיר משתנה שירוץ בלולאה.

במקרה שלנו, הפרמטר המשתנה הוא n ונגדיר עבורו משתנה i . (כל n מוחלף ב-i)

1. כל פרמטר שמשתנה ברקורסיה מגדרים עבורו לולאה (הלולאות יהיו מקוננות – סדר הלולאות צריך להיות מוגדר נכון) .
2. כל משפט return מוחלף בהשמה לטבלה ע"פ הכניסות שהוגדרו במשתנים עברו הפרמטרים. במקרה שלנו: A[i].
3. כל קריאה רקורסיבית מוחלפת בפניה לטבלה ע"פ הכניסות שמוגדרות בקריאה הרקורסיבית. במקרה שלנו: A[i-1], A[i-2].

fib(n)

A[n]

for i<- 1 to n

if n=1 or n=2

A[i] <- 1

else

A[i] = A[i-1]+ A[i-2]

return A[n]

// שיפור 1

fib(n)

A[n]

A[1] <- A[2] <- 1

for i<- 3 to n

A[i] = A[i-1]+ A[i-2]

return A[n]

//שיפור 2

fib(n)

prev1 <- prev2 <- 1

for i<- 3 to n

temp<- prev1+prev2

prev1<-prev2

prev2 <-temp

return prev2

דוגמה 2: בעית תרמיל הגב בשלמים

Wmax

n = items

Wn = weight

Vn = value

אלגוריתם רקורסיבי:

Knapsack0-1(W,V, n, Wmax)

if n=0 or Wmax=0

return 0

// אם לא ניתן לקחת את הפריט ה- n כי הוא כבד מידי

if Wn>Wmax

retrun Knapsack0-1(W,V,n-1,Wmax)

else

max( Vn+Knapsack0-1(W,V,n-1,Wmax-Wn) , Knapsack(W,V,n-1,Wmax) )

פתרון איטרטיבי:

1. נשתמש בטבלה דו מימדית (כי יש 2 פרמטרים שמשתנים בקריאות הרקורסיביות שהם n, Wmax)

גודל הטבלה לפי הטווח שלהם: n = 0-n Wmax = 0-n A[n, Wmax]

1. באיזה סדר לסרוק את הטבלה.
2. מילוי הטבלה

n=i Wmax=j

Knapsack0-1(W,V,n, Wmax)

A[n+1,Wmax+1]

for i<- 0 to n

for j<-0 to Wmax

if i=0 or j=0

A[i,j] <-0

else

if Wi > j

A[i,j] <-A[i-1, j]

else

a[i,j] <- max(Vi + A[i-1,j-Wi], A[i-1,j])

return A[n+1,Wmax+1]

הפתרון ב-#C:

static void Main(string[] args)

{

int[] w = new int[] { 2, 2, 6, 5, 4 }, v = new int[] { 2, 3, 5, 4, 6 };

int W = 10;

int[,] c = new int[w.Length + 1, W + 1];// המערך מאופס בהגדרה שלו

for (int i = 1; i < w.Length+1; i++)

{

for (int j = 1; j < W+1; j++)

{

if (w[i - 1] > j)

c[i, j] = c[i - 1, j];

else

if (v[i - 1] + c[i - 1, j - w[i - 1]] > c[i - 1, j])

c[i, j] = v[i - 1] + c[i - 1, j - w[i - 1]];

else

c[i, j] = c[i - 1, j];

}

}

Console.WriteLine(c[w.Length,W]);

// שחזור הפתרון

for (int i = w.Length, j=W; i > 0; i--)

{

if (c[i, j] != c[i - 1, j])

{

Console.WriteLine(i + " ");

j = j - w[i - 1];

}

}

}

דוגמא 3: תת סדרה המשותפת הארוכה ביותר

LCS(X,Y, n, m)

if n=0 or m=0

return 0

else

if(Xn = Ym)

return 1+LCS(X,Y, n-1,m-1)

else

return max( LCS(X,Y, n-1,m) , LCS(X,Y,n, m-1))

נעביר לפתרון תכנות דינאמי:

נגדיר טבלה דו מימדית A[n+1,m+1]

i=n j=m

ניצור לולאה שעוברת בסדר הנכון

LCS(X,Y,n,m)

A[n+1,m+1]

for i=0 to n

for j=0 to m

if i=0 or j=0

A[i,j]=0

else

if Xi= Yj

A[i,j]= 1+A[i-1,j-1]

else

A[i,j] = max( A[i-1,j], A[i,j-1] )

return A[n,m]

הקוד בשפת #C:

static void Main(string[] args)

{

string X = "ABCBDAF", Y = "BDCABA";

int x= X.Length,y=Y.Length;

int[,] m = new int[x+1, y+1]; // המערך מאופס

for (int i = 1; i < x+1; i++)

{

for (int j = 1; j < y+1; j++)

{

if (X[i - 1] == Y[j - 1])

m[i, j] = 1 + m[i - 1, j - 1];

else

m[i, j] = m[i - 1, j] > m[i, j - 1] ? m[i - 1, j] : m[i, j - 1];

}

}

Console.WriteLine(m[x,y]);

Console.WriteLine();

//שחזור - מדפיס בסדר הפוך ניתן לשמור במערך ואחכ להדפיס בסדר המתאים

for (int i = X.Length,j=Y.Length; i > 0 && j>0; )

{

if (X[i - 1] == Y[j - 1])

{

Console.Write(X[i - 1] + " ");

i--;

j--;

}

else

if (m[i, j] == m[i, j - 1])

j--;

else

i--;

}

Console.WriteLine();

}

דוגמא 4: כפל מטריצות:

const int N = 5;

int[,] m= new int[2,N]{{8,3,2,19,18},{3,2,19,18,7}}, M=new int[N,N];

int min;

for (int j = 1; j < N; j++)

{

for (int i = 0; i < N-j; i++)

{

min=int.MaxValue;

for (int k = i; k < j+i; k++)

{

if (M[i, k] + M[k + 1, j + i] + m[0, i] \* m[0, k + 1] \* m[1, j + i] < min)

min = M[i, k] + M[k + 1, j + i] + m[0, i] \* m[0, k + 1] \* m[1, j + i];

}

M[i,j+i]=min;

}

}

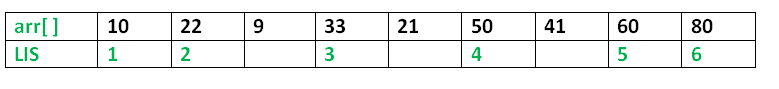
Console.WriteLine(M[0,N-1]);

Console.WriteLine();

דוגמא 5: אורך התת סדרה העולה הארוכה ביותר

Longest Increasing Subsequence | DP-3

We have discussed [Overlapping Subproblems](https://www.geeksforgeeks.org/dynamic-programming-set-1/) and [Optimal Substructure](https://www.geeksforgeeks.org/dynamic-programming-set-2-optimal-substructure-property/) properties.

Let us discuss Longest Increasing Subsequence (LIS) problem as an example problem that can be solved using Dynamic Programming.  
The Longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in increasing order. For example, the length of LIS for {10, 22, 9, 33, 21, 50, 41, 60, 80} is 6 and LIS is {10, 22, 33, 50, 60, 80}.  


More Examples:

Input : arr[] = {3, 10, 2, 1, 20}

Output : Length of LIS = 3

The longest increasing subsequence is 3, 10, 20

Input : arr[] = {3, 2}

Output : Length of LIS = 1

The longest increasing subsequences are {3} and {2}

Input : arr[] = {50, 3, 10, 7, 40, 80}

Output : Length of LIS = 4

The longest increasing subsequence is {3, 7, 40, 80}

Static int LIS(int[] arr)

{

Int[] lis= new int[arr.Length];

For(int i=0; i<lis.Length; i++)

Lis[i]=1;

For(int i=1; i<arr.Length; i++)

For(int j=0; j<i; j++)

If(arr[i]> arr[j] && lis[j]+1> lis[i])

Lis[i]= lis[j]+1;

Int max=0;

For(int i=0; i<lis.Length; i++)

If(lis[i]>max)

Max=lis[i];

Return max;

}

דוגמא 6: בעית מתן העודף

נתונות המטבעות 1, 5, 6

האלג החמדן לא יעיל לדגומא עבור סכום של 10 החמדן יחזיר 5 מטבעות בעוד שניתן להחזיר ע"י 2 מטבעות.

סכום עודף n.

Arr = 1, 2, 3, 4, 1, 1

Static int MinCoins(int n)

{

Int[] arr= new int [n+1];

Arr[1]=1; arr[2]=2; arr[3]=3; arr[4]=4; arr[5]=1; arr[6]=1;

For(i=7; i<=n; i++)

{

Arr[i]= Min(arr[i-6],arr[i-5],arr[i-1])+1

}

Return arr[n];

}

דוגמא 7: תת מערך שסכומו sum

Subset Sum Problem | DP-25

Given a set of non-negative integers, and a value *sum*, determine if there is a subset of the given set with sum equal to given *sum*.  
**Example:**

Input: set[] = {3, 34, 4, 12, 5, 2}, sum = 9

Output: True //There is a subset (4, 5) with sum 9.

[**Recommended: Please solve it on “*PRACTICE* ” first, before moving on to the solution.**](https://practice.geeksforgeeks.org/problems/subset-sum-problem/0)

Let isSubSetSum(int set[], int n, int sum) be the function to find whether there is a subset of set[] with sum equal to *sum*. n is the number of elements in set[].

The isSubsetSum problem can be divided into two subproblems  
…a) Include the last element, recur for n = n-1, sum = sum – set[n-1]  
…b) Exclude the last element, recur for n = n-1.  
If any of the above the above subproblems return true, then return true.

Following is the recursive formula for isSubsetSum() problem.

static bool isSubsetSum(int []set, int n, int sum)

    {

        // The value of subset[i][j] will be true if there

        // is a subset of set[0..j-1] with sum equal to i

        bool [,]subset = new bool[n+1,sum+1];

        // If sum is 0, then answer is true

        for (int i = 1; i <= sum; i++)

         subset[0, i] = false;

        // If sum is not 0 and set is empty, then answer is false

        for (int i = 0; i <= n; i++)

         subset[i, 0] = true;

        // Fill the subset table in bottom up manner

        for (int i = 1; i <= n; i++)

        {

            for (int j = 1; j <= sum; j++)

            {

                subset[i, j] = subset[i-1, j];

                if (set[i-1]<=j)

subset[i, j] = subset[i-1, j-set[i-1]] ;

            }

        }

for( i=n, j=sum; j>0; i--)

if(subset[i-1,j]==false)

{ c.w(set[i-1]); j-= set[i-1]; }

        return subset[sum,n];

    }

דוגמא 8 : חלוקת המערך ל-2 חלקים כך שההפרש ביניהם יהיה מינימאלי

Partition a set into two subsets such that the difference of subset sums is minimum

Given a set of integers, the task is to divide it into two sets S1 and S2 such that the absolute difference between their sums is minimum.

If there is a set S with n elements, then if we assume Subset1 has m elements, Subset2 must have n-m elements and the value of abs(sum(Subset1) – sum(Subset2)) should be minimum.

Example:

Input: arr[] = {1, 6, 11, 5}

Output: 1

Explanation:

Subset1 = {1, 5, 6}, sum of Subset1 = 12

Subset2 = {11}, sum of Subset2 = 11

static bool minPartition(int []set, int n)

    {

Int sum=0,totsum;

        for (int i = 0; i < m; i++)

         sum+=set[i];

totsum=sum;

sum/=2;

        // The value of subset[i][j] will be true if there

        // is a subset of set[0..j-1] with sum equal to i

        bool [,]subset = new bool[n+1,sum+1];

        // If sum is 0, then answer is true

        for (int i = 1; i <= sum; i++)

         subset[0, i] = false;

        // If sum is not 0 and set is empty, then answer is false

        for (int i = 0; i <= n; i++)

         subset[i, 0] = true;

        // Fill the subset table in bottom up manner

        for (int i = 1; i <= n; i++)

        {

            for (int j = 1; j <= sum; j++)

            {

                subset[i, j] = subset[i-1, j];

                if (set[i-1]<=j)

subset[i, j] = subset[i-1, j-set[i-1]] ;

            }

        }

Int res;

for( j=sum; j>0; j--)

if(subset[n,j]==true)

{ res= j;

Break; }

        return Math.Abs(res-(totsum-res));

    }

דוגמא 9: חיתוך מוט

Cutting a Rod | DP-13

Given a rod of length n inches and an array of prices that contains prices of all pieces of size smaller than n. Determine the maximum value obtainable by cutting up the rod and selling the pieces. For example, if length of the rod is 8 and the values of different pieces are given as following, then the maximum obtainable value is 22 (by cutting in two pieces of lengths 2 and 6)

length | 1 2 3 4 5 6 7 8

--------------------------------------------

price | 1 5 8 9 10 17 17 20

And if the prices are as following, then the maximum obtainable value is 24 (by cutting in eight pieces of length 1)

length | 1 2 3 4 5 6 7 8

--------------------------------------------

price | 3 5 8 9 10 17 17 20

static int cutRod(int price[],int n)

    {

        int val[] = new int[n+1];

        val[0] = 0;

        // Build the table val[] in bottom up manner and return

        // the last entry from the table

        for (int i = 1; i<=n; i++)

        {

            int max\_val = Int.MINVALUE;

            for (int j = 0; j < i; j++)

                max\_val = Math.Max(max\_val,

                                   price[j] + val[i-j-1]);

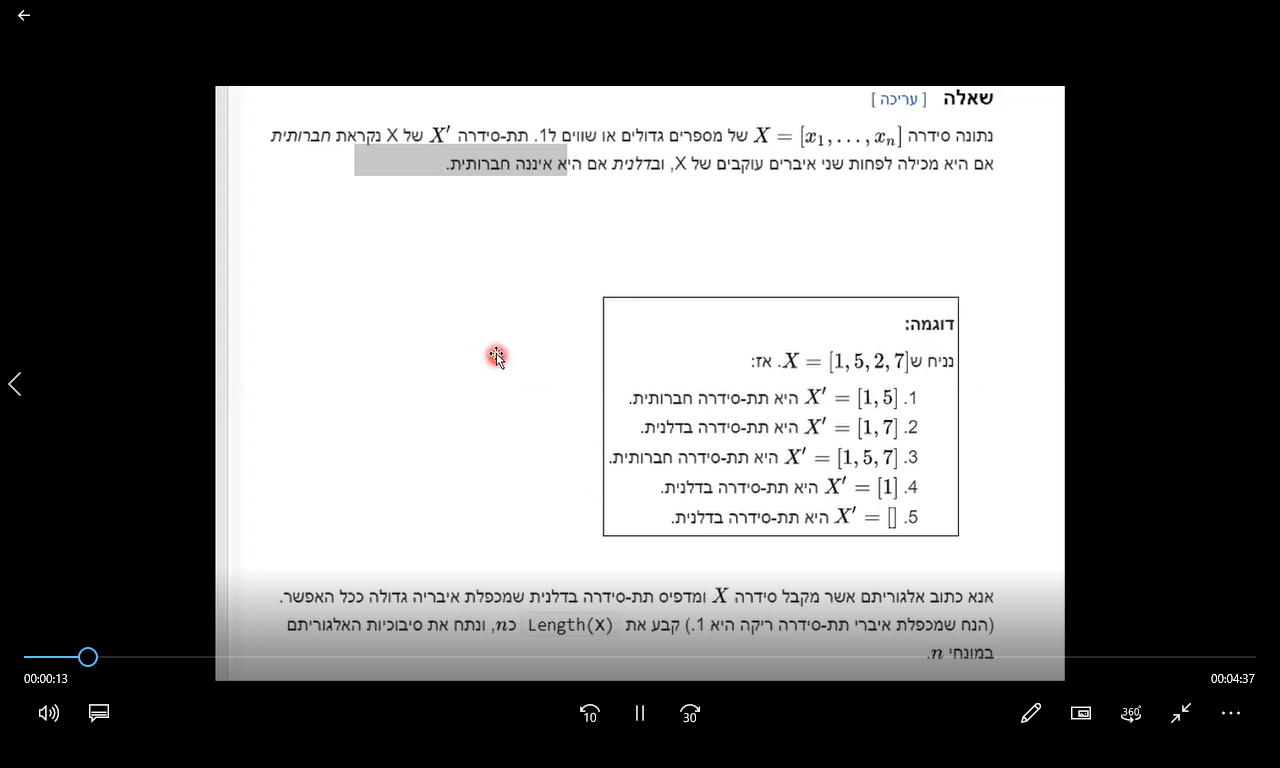
            val[i] = max\_val;

        }

        return val[n];

    }

דוגמא 10: סדרה בדלנית



A[n+1]

A=1, 1,5,5,35

A[0]=1

A[1]=X[0]

For( i=2; i<n+1; i++)

A[i]= max(A[i-1],X[i-1]\*A[i-2])

Return A[n]