

## Динамичко програмирање 2

Во наредниот текст решени се уште некои проблеми со примена на динамичкото програмирање. Користени се следните договори и ознаки:

Ако  $A$  е низа од  $N$  елементи, со  $A_k$  ја означуваме низата која што се состои од првите  $k$  елементи на низата  $A$ , каде  $k \in N$ . Притоа,  $A_0$  е празната низа, а  $A_N$  е еднаква на целата низа  $A$ . Елементите на низата  $A$  ги означуваме со  $a_1, a_2, \dots, a_N$  или со  $A(1), A(2), \dots, A(N)$ . Во случај на двојни индекси, поради подобра читливост нема да пишуваме туку  $a(b_c)$ , или  $a(b(c))$ . Истите забелешки важат и за матриците.

### 1) Проблем на максимален збир

Нека е дадена правоаголна шема  $A$  со  $M \times N$  полиња сместени во  $M$  редици и  $N$  колони и пополнети со цели броеви. Од секое поле е дозволено да се премине само на полето под или на полето десно од тоа поле. Потребно е да се избере пат од горното лево поле до долното десно поле, така што збирот на броевите во полињата преку кои се поминува треба да биде максимален. Да се испише вредноста на оптималниот пат, а потоа и патот како низа на координатни полиња преку кои се поминува.

#### Решение:

Како што може да се претпостави, генерирањето на сите можни патишта и памтење на оној пат кој што има најголем збир, не е добра идеја, бидејќи бројот на можни патишта расте со експоненцијална брзина со порастот на големината на правоаголникот.

Да се потсетиме на условите за примената на динамичко програмирање:

- **оптималното решение на проблемот содржи во себе оптимални решенија на проблемите од истиот тип содржани во главниот проблем**
- **потпроблемите имаат заеднички потпроблеми, односно делумно се преклопуваат**

Да ги провериме условите за примена на динамичкото програмирање. Нека е даден патот  $P$  чии што полиња имаат најголем збир. Тогаш секој дел од оптималниот пат го соединува почетното и крајното поле на тој дел на оптимален начин (инаку почетниот пат не би бил оптимален). Згодно е потпроблемите формално да ги зададеме на следниот начин: нека  $P(i,j)$  е оптималниот пат од полето  $(1,1)$  до полето  $(i,j)$  и нека  $B(i,j)$  е збирот кој што се постигнува на тој пат. Тогаш се бара патот  $P(M,N)$  и збирот  $B(M,N)$ .

Веќе заклучивме дека проблемот има оптимална потструктура. Можеме ли  $B(M,N)$  да го изразиме рекурентно? До полињата  $(M,N)$  можеме да дојдеме само преку едно од полињата  $(M,N-1)$  или  $(M-1,N)$ . Затоа збирот  $B(M,N)$  е еднаков на поголемиот од вредноста  $B(M,N-1)$  или вредноста  $B(M-1,N)$  зголемена за  $A(M,N)$  односно:

$$B(M,N) = \max\{B(M,N-1), B(M-1,N)\} + A(M,N)$$

Тогаш патот  $P(M,N)$  се добива со додавање на полето  $(M,N)$  на оној пат  $P(M-1,N)$  или  $P(M,N-1)$  кој што дава поголем збир.

1,1	→ 1,2	....	1,N
2,1	↓	2,2	.... 2,N
3,1	3,2	→ ....	3,N
....	....	....	....
M,1	M,2	→ ...	M,N

1,1			
			M-1,N
		M,N-1	M,N

Според тоа, решавањето на почетниот проблем се сведува на решавање на два проблема од ист тип и нивно едноставно комбинирање односно споредување на два збира и собирање на поголемиот од тие два збира со вредноста на последното поле. За решавање на сите нетривијални потпроблеми важи потполно истото.

Тривијалните проблеми се наоѓање на елементи од првиот ред и првата колона на матрицата  $B$ . Бидејќи до полињата  $(1, Y)$  или  $(X, 1)$  постои само еден пат, треба само да се соберат полињата на тој пат.

Потпроблемите и кај овој проблем се поклопуваат. На пример проблемот за полето  $(M-1, N-1)$  се појавува и кај потпроблемот  $(M, N-1)$  и кај  $(M-1, N)$ . Затоа за решавање на овој проблем никако не е прифатлива рекурзијата.

Повеќе потпроблеми со рекурзивно решение се сведуваат на два помали проблеми, па според тоа со рекурзија потребно е експоненцијално да се решат многу потпроблеми наместо само  $M \cdot N$ , колку вкупно ги има различни. Затоа, по ред ќе ги решиме сите различни потпроблеми, т.е. ќе го користиме динамичкото програмирање.

Координатите на полињата преку кои минува оптималниот пат ги пронаоѓаме од последното кон првото поле. За да ги испишеме од првото кон последното, може да користиме стек, со помош на кој редоследот на податоците го вртиме наопаку.

Меѓутоа, при повикување на потпрограмите, оперативниот систем веќе користи стек за сместување на потребните податоци, па примената на рекурзијата е едноставен и природен начин за да се испишат координатите на полињата во саканиот редослед.

**Забелешка:** Решението е добиено со динамичко програмирање, а рекурзија се користи само за испишување на решението.

Потпрограмата  $ispis(M, N)$  која што го испишува патот до полето  $(M, N)$ , се повикува рекурзивно најмногу онолку пати колку што има полиња на патот од полето  $(1, 1)$  до полето  $(M, N)$ , односно вкупно помалку од  $M + N$  пати, така што таа бргу се извршува.

```

var A,B:array[1..30,1..30]of integer;
i,j,M,N:integer;
procedure ispisi(x,y:integer);
begin
  if(x>1)and(y>1)then
    begin
      if B[x-1,y]>B[x,y-1] then ispisi(x-1,y)
      else ispisi(x,y-1);
      writeln(x,' ',y);
    end
  else
    if(x=1) then
      for i:=1 to y do writeln(x,' ',i)
    else
      for i:=1 to x do
        writeln(i,' ',y);
      end;
  end;
Begin
  writeln('M N');
  readln(M,N);
  for i:=1 to M do
    for j:=1 to n do
      begin
        write('A[' ,i ,', ',j ,']=');
        readln(A[i,j]);
      end;
  writeln('Matricata A e ');
  for i:=1 to M do
    begin
      for j:=1 to n do
        write(A[i,j]:3);
      writeln;
    end;
  B[1,1]:=A[1,1];
  for j:=2 to N do B[1,j]:=A[1,j]+B[1,j-1];
  for i:=2 to M do B[i,1]:=A[i,1]+B[i-1,1];
  for i:=2 to M do
    for j:=2 to N do
      if B[i,j-1]<B[i-1,j] then B[i,j]:=A[i,j]+B[i-1,j]
      else B[i,j]:=A[i,j]+B[i,j-1];
  writeln('Maksimalniot zbir e ',B[M,N],' a se postignuva vaka:');
  ispisi(M,N);
end.

```

Оваа задача може да се појави во повеќе сродни варијанти. Во секоја варијанта ќе подразбираме (ако на друг начин не се напомене), дека е потребно со движење низ матрицата да се постигне најголем збир и на секое поле да може да се застане само еден пат.

- Со движење надолу и десно да се стигне од полето (1,1) до полето (m,n). Ова е варијанта која тука се разгледува и решава.
- Со движење доле, десно и доле-десно се стигнува од полето (1,1) до полето (m,n). Во оваа варијанта (нетривијална) елементите од матрицата B се сметаат незначително на друг начин:

$$B(X,Y) = \max\{B(X,Y-1), B(X-1,Y), B(X-1,Y-1)\} + A(X,Y).$$

Во склад со оваа измена, треба да ја измениме и постапката за реконструкција на решението како и самата програма.

- Со движење горе-десно, доле-десно и десно се стигнува од кое било поле (x<sub>1</sub>,1) од првата колона до кое било поле (x<sub>2</sub>,n) од последната колона.

Тука матрицата  $B$  мора да се пополнува по колони, за елементите од првата колона важи  $B(X,1)=A(X,1)$ , а за останатите

$$B(X,Y) = \max\{B(X-1,Y-1), B(X,Y-1), B(X+1,Y-1)\} + A(X,Y)$$

при што сметаме дека  $B(X-1,0)$  и  $B(X-1,M+1)$  се еднакви на  $-\infty$ , т.е. за првите и последните елементи во колоната на максимум се бира од 2, а не од 3 члена.

- Од едно поле на матрицата се преминува на следното со движење како шаховскиот коњ, но само во десно. Треба да се стигне од првата до последната колона. Задачата се решава слично како и претходните.
- Даден е триаголник пополнет со броевите: во првиот ред еден број, во вториот два броја, итн. до  $N$ -от ред со  $N$  броеви. Треба да стигнеме од бројот на врвот до бројот на основата на триаголникот. Триаголникот може да се смести во квадратна матрица на повеќе начини. Нека биде тоа триаголникот над споредната дијагонала. Така добиваме еквивалентна постапка: во квадратната матрица треба со движење надолу и десно да се стигне од горниот лев агол до кој било број на споредната дијагонала. Задачата се решава така што на ист начин како и порано. Формираме квадратна матрица  $B$ , но ја пополнуваме само до споредната дијагонала. Решението е одредено со најголемите елементи на споредната дијагонала на матрицата  $B$ .
- Целта е со движење надолу и десно да се стигне од горното лево до долното десно поле на матрицата  $A$ , при тоа да се застане на такви соседни броеви  $x_1, x_2, \dots, x_k$  во матрицата  $A$  ( $x_1=a_{11}$ ,  $x_k=a_{mn}$ ), така што ќе се максимизира не нивниот збир, туку збирот на апсолутните разлики на

непрекинатите броеви преку кои се поминува, односно вредноста  $\sum_{i=1}^{k-1} |x_{i+1} - x_i|$ .

Лесно се воочува дека за  $X > 1$ ,  $Y > 1$  важи релацијата

$$B(X,Y) = \max \left\{ \begin{array}{l} B(X,Y-1) + |A(X,Y-1) - A(X,Y)| \\ B(X-1,Y) + |A(X-1,Y) - A(X,Y)| \end{array} \right.$$

додека елементите од првата редица и првата колона се пресметуваат според формулата

$$B(1,1) = 0$$

$$B(1,Y) = B(1,Y-1) + |A(1,Y-1) - A(1,Y)|$$

$$B(X,1) = B(X-1,1) + |A(X-1,1) - A(X,1)|$$

Врз основа на вредностите на пополнетата матрица  $B$  не е тешко да се одредат полињата низ кои се поминува поради максимизирање на бараниот збир.

- Треба да се стигне од кој било елемент на првата колона до кој било елемент на последната колона, движејќи се горе, доле и десно.

Матрицата  $B$  ја пополнува по колони така што  $B(X,Y)$  да е најголема вредност која може да се достигне со движење низ првите  $Y$  колони и застанување во полето  $A(X,Y)$ . Тогаш имаме:

$$B(X,1) = \max_{1 \leq k \leq M} \left\{ \sum_{p=k, X}^1 a_{p,1} \right\}, \quad 1 \leq X \leq M$$

каде  $p$  се менува од  $k$  до  $X$ , а за  $k > X$  оди и наназад, и

$$B(X, Y) = \max_{k \leq X \leq M} \left\{ B(k, Y-1) + \sum_{p=k, X}^Y \alpha_{p, Y} \right\}, \quad 1 \leq X \leq M, \quad 2 \leq Y \leq N$$

Вкупниот број на операции во овој пример е поголем и е пропорционален со  $M^2N$ , а резултатот е одреден со максимумот на последната колона од матрицата В.

Освен наведените, може да се најдат и други слични варијанти на овој проблем. Сите тие се решаваат на начин многу близок на изложениот.

## 2) Најдолга заедничка подниза

Низата А е подниза на низата В, ако со прецртување на некои елементи од низата В може да се добие низата А. На пример (1,3,3,5) е подниза од (1,2,3,3,4,5), а не е подниза ниту од (1,5,2,3,3,4), ниту од (1,2,3,4,5).

Дадени се две низи: P од M и Q од N елементи. Да се најде низата со најголема можна должина која е подниза и за P и за Q.

### Решение:

Нека  $NZP(X, Y)$  е најдолгата заедничка подниза од низите  $P_X$  и  $Q_Y$ . Во задачата се бара  $NZP(M, N)$ . Ако е  $p_M = q_N$ , тогаш  $NZP(M, N) = NZP(M-1, N-1) \cup \{p_M\}$ , ( $p_M$  се допишува на крајот од низата  $NZP(M-1, N-1)$ ), додека за  $p_M \neq q_N$ ,  $NZP(M, N)$  е еднаков на подолгиот пат од  $NZP(M-1, N), NZP(M, N-1)$ .

Ова релација ни овозможува да ги пресметаме сите  $NZP(X, Y)$  по ред (по редици или по колони), знаејќи дека е  $NZP(X, 0) = NZP(0, Y) = \emptyset$  (празна низа). Доволно е да се паметат должините на најдолгите заеднички поднизи во матрицата В, а  $NZP(X, Y)$  лесно се реконструира на основа на матрицата В.

$$B(X, 0) = B(0, Y) = 0$$

$$B(X, Y) = \begin{cases} B(X-1, Y-1) + 1, & \text{ako e } P[X] = Q[Y] \\ \max\{B(X, Y-1), B(X-1, Y)\}, & \text{ako e } P[X] \neq Q[Y] \end{cases}$$

Доколку се бара само должината  $NZP$ , можеме да заштедиме простор, така што наместо матрицата В користиме само две низи, кои играат улога на редица од В која моментално се формира и претходната редица (со редослед на пополнување наназад, доволна е само една низа).

За да ја реконструираме  $NZP$  неопходна ни е целата матрица В, или дополнително време за повторно пресметување на изгубените информации.

```

var
    B:array[0..100,0..100] of integer
    p,q:array[1..100] of integer;
    i,j,m,n:integer;

procedure ispisi(i,j:integer);
begin
    if(i>0) and (j>0) then
        if p[i]=q[j] then begin
            ispisi(i-1,j-1);
            write(p[i]:3);
        end
        else if B[i-1,j]>B[i,j-1] then
            ispisi(i-1,j)
        else
            ispisi(i,j-1);
    end;
begin
    {vnesuvanje na nizata P}
    write('m='); readln(m);
    for i:=1 to m do begin
        write('p[' ,i,']= ');
        readln(p[i]);
    end;

    {vnesuvanje na nizata Q}
    write('n='); readln(n);
    for j:=1 to n do begin
        write('q[' ,j,']= ');
        readln(q[j]);
    end;

    for i:=0 to m do B[i,0]:=0;
    for j:=0 to n do B[0,j]:=0;
    for i:=1 to m do
        for j:=1 to n do
            if p[i]=q[j] then
                B[i,j]:=B[i-1,j-1]+1
            else if B[i-1,j] > B[i,j-1] then

```

### **Пример :**

Нека се дадени низите  $P=\{1,3,4\}$  и  $Q=\{2,1,5,4,3\}$ .

Да се најде најдолгата подниза на P и на Q.

### **Решение :**

$M=3, N=5;$

Ја пополнуваме матрицата B која е од ред  $(M+1) \times (N+1)$

Нултата редица и нултата колона ја полниме со нули.

Во контекст на програмското решение матрицата B ја полниме на следниот начин:

$$P[1]=1 \neq Q[1]=2 \Rightarrow B(1,1)=\max\{B(1,0), B(0,1)\}=0$$

$$P[1]=1=Q[2]=1 \Rightarrow B(1,2)=B(0,1)+1=1$$

0	0	0	0	0	0
0	0	1			
0					
0					

0		0	0	0	0
0		0			
0					
0					

$$P[1]=1 \neq Q[3]=5 \Rightarrow B(1,3)=\max\{B(1,2), B(0,3)\}=1 \dots$$

$$P[2]=3 \neq Q[1]=2 \Rightarrow B(2,1)=B(2,0)=0 \dots$$

0	0	0	0	0	0
0	0	1	1		
0					
0					

0	0	0	0	0	0
0	0	1	1	1	1
0	0				
0					

$$P[2]=3 \neq Q[2]=1 \Rightarrow B(2,2)=\max\{B(1,2), B(2,1)\}=1 \dots$$

$$P[2]=3 = Q[5]=3 \Rightarrow B(2,5)=B(1,4)+1=2$$

0	0	0	0	0	0
0	0	1	1	1	1
0	0	1	1	1	2
0					

0	0	0	0	0	0
0	0	1	1	1	1
0	0	1			
0					

$$P[3]=4 \neq Q[1]=2 \Rightarrow B(3,1)=\max\{B(2,1), B(3,0)\}=0 \dots$$

$$P[3]=4 = Q[4]=4 \Rightarrow B(3,4)=B(2,3)+1=2 \dots$$

0	0	0	0	0	0
0	0	1	1	1	1
0	0	1	1	1	2
0	0				

0	0	0	0	0	0
0	0	1	1	1	1
0	0	1	1	1	2
0	0	1	2	2	2

Изгледот на поднизата се наоѓа со помош на матрицата  $B$  почнувајќи од последното поле. Се испитуваат условите зададени во процедурата  $ispisi(i,j)$  за да се увиди во која насока да треба да се продолжи со повикувањето на процедурата.

$ispisi(i,j); i=3,j=5;$

$P[3]=4 \neq Q[5]=4 \Rightarrow$  од не  $(B(2,5)=2 > B(3,4)=2) \Rightarrow ispisi(3,4)$

$P[3]=4 = Q[4]=4 \Rightarrow ispisi(2,3)$  печати  $P[3]$

$P[2]=3 \neq Q[3]=5 \Rightarrow$  од не  $(B(1,3)=1 > B(2,2)=1) \Rightarrow ispisi(2,2)$

$P[2]=3 \neq Q[2]=1 \Rightarrow$  од да  $(B(1,2)=1 > B(2,1)=0) \Rightarrow ispisi(1,2)$

$P[1]=1 = Q[2]=1 \Rightarrow ispisi(0,1)$  печати  $P[1]$

Завршивме затоа што  $i=0$

Значи низата што се добива е  $\{1,4\}$

### 3) Најефтина исправка на зборови

Дозволените операции над стрингот се: вметнување букви, бришење на една буква, измена на буквите и бришење на сите букви до крајот на стрингот. Секоја од овие операции има зададена цена.

Потребно е да се одреди најмалата вкупна цена на операциите со кои од даден стринг  $A$  се добива дадениот стринг  $B$ .

#### Решение:

Задачата се решава многу слично како проблемот на максималниот збир и проблемот за најдолга заедничка подниза, па затоа деталното објаснување е изоставено.



```

var
    cena: array[0..50,0..50] of real;
    a,b: string;
    cVmet, cBris1, cZam, cBrisDoKraj:real;
    i,j,k,m,n:integer;

function min3(a,b,c:real):real;
var x:real;
begin
    x:=a; if x>b then x:=b; if x>c then x:=c; min3:=x;
end;

begin
    write('Cena na vmetnuvanja='); readln(cVmet);
    write('Cena na brisenje na eden znak='); readln(cBris1);
    write('Cena na zamena='); readln(cZam);
    write('Cena na brisenje do kraj na stringot='); readln(cBrisdoKraj);
    write('a= '); readln(a); m:=length(a);
    write('b= '); readln(b); n:=length(b);
    cena[0,0]:=0;
    for i:=1 to m do begin
        cena[i,0]:=cena[i-1,0]+cBris1;
        if cena[i,0]>cBrisDoKraj then
            cena[i,0]:=cBrisDoKraj;
    end;
    for j:=1 to n do
        cena[0,j]:=cena[0,j-1]+cVmet;
        for i:=1 to m do
            for j:=1 to n do begin
                if a[i] = b[j] then
                    cena[i,j]:=cena[i-1,j-1]
                else
                    cena[i,j]:=min3(cena[i-1,j-1]+cZam,
                                     cena[i,j-1] + cVmet,cena[i-1,j]
+ cBris1);
                for k:=1 to i-1 do
                    if cena[i,j]>cena[k,j]+cBrisDoKraj then
                        cena[i,j]:=cena[k,j]+cBrisDpKraj;
                end;
            writeln('najeftinata ispravka cini',cena[m,n]:10:3);
    end.

```

#### 4) Максимална сума на несоседни во низа

Дадена е низата  $A$  од  $N$  цели броеви. Да се одреди поднизата  $A$  чиј збир на елементите е максимален, а во кој нема соседни елементи на низата  $A$ . Да се смета дека празната низа има збир на елементите нула.

#### Решение:

Нека за низата  $A$  е позната една таква подница  $P=P(A)$ . Ако елементот  $a_N$  не припаѓа на поднизата  $P$ , тогаш  $P$  е оптимална подница и за низата  $A_{N-1}$ . Ако елементот  $a_N$  припаѓа на поднизата  $P$ , тогаш поднизата  $P$  е без последен елемент (т.е. без  $a_N$ ). Во спротивно би било можно да се подобри оптималната подница  $P$  на низата  $A$  така што на подобрата подница на низата  $A_{N-2}$  ќе се допише бројот  $a_N$ . Ја установивме оптималноста на подструктурата на проблемот.

Решенијата за низите  $A_0$  и  $A_1$  се тривијални. За  $X$  од 2 до  $N$ ,  $P(A_x)$  е оној елемент од низите  $P(A_{x-1})$  и  $P(A_{x-2}) \cup \{a_x\}$ , кој што има поголем збир.

Ќе формираме низа  $B$ , така што е  $B(X)$  збир на елементите на оптималната подница  $P(A_x)$ . Според претходното следува

$$B(0) = 0, B(1) = \max(0, a_1), B(X) = \max\{B(X-1), B(X-2)+a_x\}, X=2, N.$$

На основа на низата  $B$  едноставно се наоѓа поднизата  $P$ .

```
var
    a,b:array[0..1000] of integer;
    n,i:integer;

procedure ispis(n:integer);
begin
    if n>0 then
        if b[n]=b[n-1] then
            ispis(n-1)
        else begin
            ispis(n-2);
            write(a[n]:4);
        end;
    end;
end;

begin
    readln(n);
    for i:=1 to n do readln(a[i]);
    b[0]:=0;
    if a[1]>0 then b[1]:=a[1] else b[1]:=0;

    for i:=2 to n do
        if b[i-2]+a[i]>b[i-1] then
            b[i]:=b[i-2]+a[i]
        else
            b[i]:=b[i-1];

    ispis(n); readln;
end.
```

### Пример:

Нека е дадена низата  $A$  со  $n=6$  елементи.  $A=\{4,5,6,1,8,9\}$

### Решение:

Ја пополнуваме низата  $B$  со помош на формулите зададени претходно

```
B[0]=0; B[1]=A[1]=4;

B[2]=max{B[0]+A[2], B[1]} =max{5,4}=5

B[3]= max{B[1]+A[3],B[2]}=max{10,5}=10

B[4]=max{B[2]+A[4],B[3]}=max{6,10}=10

B[5]=max{B[3]+A[5],B[4]}=max{18,10}=18

B[6]=max{B[4]+A[6],B[5]}=max{19,18}=19
```

За да ја определеме поднизата на низата  $A$  почнуваме од крајот на низата  $B$  одејќи на напред проверувајќи ги условите во процедурата  $ispis(n)$ ;

```
Ispis(6); n=6;
```

```
B[6]=19 ≠ B[5]=18 => ispis (6-2) , печати A[6]=9
```

```
B[4]=10 = B[3]=10 => ispis (4-1)
```

```
B[3] =10 ≠ B[2] =5 => ispis (3-2), печати A[3]=6
```

```
B[1]=4 ≠ B[0]=0 => ispis(1-2), печати A[1]=4
```

Завршивме затоа што се доби  $n < 0$

Поднизата со максимален збир на дадената низа A е {4,6,9}