



BLM3021
ALGORİTMA ANALİZİ

ÖDEV-3

DİNAMİK PROGRAMLAMA
PROBLEM-1

Problem1: A ve B takımları, içlerinden biri n galibiyet alana kadar maç yapacaklardır. A takımının bir maçı kazanma olasılığı her maç için p, kaybetme olasılığı ise 1 p'dir. Dolayısıyla beraberlik ihtimali mevcut değildir. A takımının seriyi kazanmak için i tane daha maç kazanması, B takımının da seriyi kazanmak için j tane maç kazanması gereken durumda A'nin seriyi kazanma olasılığı $P(i,j)$ 'dir.

- $P(i,j)$ 'yi kullanarak **dinamik programlama** ile algoritma tasarlayabilmek için **rekürans bağıntısını** yazınız.
- A takımının bir maçı kazanma olasılığının 0.6 olduğu durumda 7 maçlık seride (4 alan kazanır) **A takımının kazanma ihtimalini** hesaplayınız. (Dinamik programlama yaklaşımını kullanınız.)
- Belirtilen problemi çözen dinamik programlama yaklaşımını kullanan algoritmanın **kodunu C dilinde** yazınız.

ÇÖZÜM

Programın akışı ve çalışma yapısı:

Verilen bu problemin çözümünü karşılayan programda A takımının kazanma olasılığı göz önünde tutularak; kullanıcıdan A ve B takımları arasında gerçekleşecek olan toplam maç sayısını ve A takımının kazanma olasılığını(P) girilmesi istenir.

Bu girdilerden sonra programın işleyişi başlar ve ilk olarak girilen P değerinin 0 ile 1 arasında olup olmadığı kontrol edilir. Gerekli kontrolden sonra programın ana akışına geçilir ve bir adet **winCount** adında değişken oluşturulur. Bu değişkende A takımının seriyi kazanması için gerekli maç sayısı tutulur. Örnek; 9 Maç girildiyse $9/2=4.5 \rightarrow 4 + 1$ işlemi yapılarak serideki toplam maçın yarısının 1 fazlası değeri verilir, böylelikle iki takımdan birisi bu kazanma sayısına ulaştığı vakit seriyi kazanmış olur.

Daha sonrasında ise bu kazanma sayısının birer fazlası değerlere sahip olan satır ve sütun sayıları ile bir matrisimiz oluşturulur. 1 fazlası olmasının sebebi ise bu tabloda iki takımın kazanmak için 0 maç ihtiyacı değerlerindeki tutulmasıdır. Örneğimizden devam edersek, $\text{winCount}=5 + 1 \Rightarrow 6$, 6x6 lık matrisimizde A takımının kazanması ve kaybetmesi için kalan maçları 0,1,2,3,4,5 şeklinde tutularak 6 eleman ortaya çıkmış olur. Matrisimizde sadece A takımının kazanması veya kaybetmesi ile ilgilenilmektedir.

Kazanmak için kalan maçları tutacak matrisimizin ilk elemanları manuel olarak initialize edilir bunun sebebi ise aşağıdaki tabloda açıklanmıştır;

A/B	0	1	2	3	4	5
0	0	1	1	1	1	1
1	0	P(A)				
2	0		P(A)			
3	0			P(A)		
4	0				P(A)	
5	0					P(A)

Kırmızı değerler=Bahsettiğimiz manuel olarak initialize edilen değerlerdir ve bunun anlamı ise; 1. Sütun 1. satır (Dış katmanlar hariç)=A takımının kazanması için gerekli kalan maç sayısı=0, B takımının ise=0. Bu ihtimal imkansız (iki takımında aynı anda) olacağından burası 0'dır. Bundan hariç 1. satırın diğer tüm değerlerinde ise A takımının kazanması gereken kalan maç sayısı 0 olduğu taktirde, A takımı seriyi her türlü kazanmış demektir ve B takımının kalan maçlarına bakılmaksızın A takımının kazanması ihtimalleri 1 olur. 1. Sütunda ise aynı şey B takımı için geçerli. Yani, B takımının kazanması için gerekli kalan maç sayısı 0 ise B takımı kazanmış demektir ve A takımının kalan kazanması gereken maç sayılarına bakılmaksızın **A takımının kazanma ihtimali 0 olur**. Bu kısımlar manuel olarak doldurulduktan sonra tablomuzun geri kalan kısmını doldurmamız için bir rekürans bağıntısı bulmamız gerekicek. Bu bağıntı ile tablomuzun (matrisimizin) geri kalan kısımlarını doldurarak A takımının seriyi kazanma olasılığını bularak programın çözümünü tamamlamış oluruz.

Rekürans bağıntısı:

Rekürans Bağıntısı Makalama

A/B	0	1	2
0	0	1	1
1	0	0,5	(x)
2	0	(y)	0,5

$P(A) = 0.5$ olsun,

$\rightarrow A, 1$ maç kazanmalı $= P(A)$

$B, 2$ maç kazanmalı $= P(B)$

$\rightarrow A, 2$ maç kazanmalı

$B, 1$ maç kazanmalı

1) $x = A / BA$

$$x = P(A) + P(B)' * P(A)' = 0.5 + (0.5)' * (0.5)'$$

$$x = 0.75$$

2) $y = B / AB$

$$y = P(B) + P(A)' * P(B)' = 0.5 + (0.5)' * (0.5)'$$

$$y = 0.75 = P(B)$$

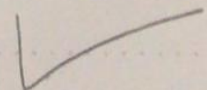
Böylelikle $P(A) = 1 - P(B) = 0.25$

A/B	0	1	2
0	0	1	1
1	0	$P(A)$	$x = 0.75$
2	0	$y = 0.25$	$P(A)$

serimik (Bağıntımcı)

$$x = P(A) * (\text{üst hücre}) + (\text{sol hücre}) * P(B)$$

$$0.5 * 1 + 0.5 * 0.5 = 0.75$$



Kağıttaki bağıntıyı yakalamak için n (Toplam maç sayısı)=3 ve $P(A)=0.5$ değerleri için verilen örnekte görüldüğü gibi tabloda ilgili alanlar doldurulup rekürans bağıntısı elde edilmiştir. Sonuç olarak programın akışındaki devamlılıkta ise 0.satırlar ve 0.sütunların manuel doldurulmasından sonra satır ve sütun indislerimiz 1den başlayarak **winCount** değişkenine kadar gittiğini göz önünde bulundurursak **rekürans bağıntımız** = $(aP * \text{matrix}[i-1][j]) + ((1-aP) * \text{matrix}[i][j-1])$. Burdaki aP değişkenimiz = $P(A)$.

Son olarak kullanıcının olasılık tablosunu görmesi amacıyla tablo ve A takımının seriyi kazanma olasılığı ekrana yazdırılarak program sonlandırılır.

Değişkenler ve tanımları:

int n=Kaç maç olacağını tutan değişkenimiz.

int winCount=Takımların maç sayısına göre kazanması için gerekli maç sayısını tutan değişkenimiz.

float aP=A takımının kazanma olasılığını tutan değişkenimiz.

float matrix[winCount+1][winCount+1]=Olasılıklar tablomuz.

Kodlar:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(int argc, char *argv[]) {
    int n, winCount;
    float aP; //A nin kazanma olasiligini tutacak olan degiskenimiz. Boylelikle B nin
    kazanma olasiligi ise = (1-aP) olur.
    printf("Kac mac yapilacagini girin:"); //kac mac yapilacagi kullanicidan
    isteniliyor.
    scanf("%d", &n);
    printf("A takiminin kazanma olasiligini (P) girin:"); //A takimi icin esit
    durumlardaki bakilacak P(gelme olasiligi) degeri. 1den buyuk deger girilmemesi
    gerekiyor.
    scanf("%f", &aP);

    if(aP >= 0 && aP <= 1) { //olasilik degeri yalnızca 0-1 arasında ise program calismaya
    devam eder.
        winCount = n / 2 + 1;
        int i, j; //matrisimizde satir ve sutun indisleri olarak kullanicagimiz
        degiskenlerimiz.

        float matrix[winCount+1][winCount+1]; //matrisimizin satir ve sutun sayilari
        kazanmak icin gerekli olan mac sayisindan 1 fazla olmalı.
        //Bunun nedeni ise kazanmak icin gerekli olan mac sayisinda 0 degerinde
        hesaplanmasidir. (nek:n=4 icin, 0->1->2->3->4=5 indis)

        for(i=0; i<winCount; i++) {
            matrix[i][0] = 0; //B takiminin kazanmak icin kalan mac sayisi 0 oldugu
            durumlarda, A takiminin kazanma ihtimali 0dir. Matrisin bu gozlerini 0 olarak
            dolduruyoruz.
            matrix[0][i+1] = 1; //A takiminin kazanmak icin kalan mac sayisi 1 oldugu
            durumlarda, B takiminin kaybetme(A takiminin kazanma)olasiligi 1dir. Satirlari bu sekilde
            dolduruyoruz.
        }
        //Elde ettigimiz baginti ile matrisin kalan yerlerini dolduruyoruz. 1.satir ve
        1.sutun komple dolduruldugu icin o hucreleri atliyoruz.
        for(i=1; i<=winCount; i++) {
            for(j=1; j<=winCount; j++) { //ilgili hucredeki kazanma olasiligini mevcut
            duruma gore hesaplamamiz icin ortaya cikardigimiz rekürans bagintimizi kullaniyoruz.
            //P(i,j) = pP(i-1,j) + (1-p)P(i,j-1)
            matrix[i][j] = (aP * matrix[i-1][j]) + ((1-aP) * matrix[i][j-1]);
        }
    }
}
```

```

        for(i=0;i<=winCount;i++){//olasiliklar tablosunu kullanicilarin gormesi icin
matrisimizi yazdiriyoruz.
            for(j=0;j<=winCount;j++){
                printf("%.2f    ",matrix[i][j]);
            }
            printf("\n");
        }
        printf("A takiminin kazanma olasiligi=%.2f",matrix[winCount][winCount]);//A
takimi icin kazanma olasiligini yazdiriyoruz.
    }
    else
        printf("Olasilik degerini minimum 0,maximum 1 girmelisiniz.");

    return 0;
}

```

Ekran çıktıları:

1)Açıklamadaki örnek.

```

Kac mac yapilacagini girin:3
A takiminin kazanma olasiligini(P) girin:0.5
0.00    1.00    1.00
0.00    0.50    0.75
0.00    0.25    0.50
A takiminin kazanma olasiligi=0.50
-----
Process exited after 2.874 seconds with return value 0
Press any key to continue . . .

```

2)Bizden istenilen durum(B.şıkkı).

```

Kac mac yapilacagini girin:7
A takiminin kazanma olasiligini(P) girin:0.6
0.00    1.00    1.00    1.00    1.00
0.00    0.60    0.84    0.94    0.97
0.00    0.36    0.65    0.82    0.91
0.00    0.22    0.48    0.68    0.82
0.00    0.13    0.34    0.54    0.71
A takiminin kazanma olasiligi=0.71
-----
Process exited after 13.1 seconds with return value 0
Press any key to continue . . .

```

3)İpucunun testi.

```
Kac mac yapılacağını girin:7
A takımının kazanma olasılığını(P) girin:0.4
0.00    1.00    1.00    1.00    1.00
0.00    0.40    0.64    0.78    0.87
0.00    0.16    0.35    0.52    0.66
0.00    0.06    0.18    0.32    0.46
0.00    0.03    0.09    0.18    0.29
A takımının kazanma olasılığı=0.29
-----
Process exited after 4.717 seconds with return value 0
Press any key to continue . . .
```

4)Programa girilmeyen durumlar.

```
Kac mac yapılacağını girin:5
A takımının kazanma olasılığını(P) girin:24
Olasılık değerini minimum 0,maximum 1 girmelisiniz.
-----
Process exited after 6.082 seconds with return value 0
Press any key to continue . . .
```

```
Kac mac yapılacağını girin:4
A takımının kazanma olasılığını(P) girin:-1
Olasılık değerini minimum 0,maximum 1 girmelisiniz.
-----
Process exited after 2.533 seconds with return value 0
Press any key to continue . . .
```