

Ders Adı: Algoritma Analizi

Öğrenci Bilgileri: Ömer Buğrahan Çalışkan – 17011076

Ödev İçeriği: 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.

Soru: 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.)

Cevap: Sorunun cevabını dinamik programlama yaklaşımı ile 0.710208 -> 0.71 olarak buldum. Cevabı bulduğum matris ve ekran görüntüsü Ekran Görüntüleri kısmına eklenmiştir.

Rekürans Bağıntısı:

- if (i= 0 ve j=0) -> $matrix[i][j] = 1$
- if (j=0) -> $matrix[i][j] = matrix[i-1][j]*(prob\ of\ A)$
- if (i=0) -> $matrix[i][j] = matrix[i][j-1]*(1-prob\ of\ A)$
- Else ->
 $matrix[i][j] =$
 $(matrix[i][0]*matrix[0][j]) *$
 $(fakt(i+j-1) / (fakt(i-1)*fakt(j)))$ then
 $matrix[i][j] = matrix[i][j] + matrix[i][j-1]$

prob of A = A takımının kazanma olasılığı.
fakt = faktöriyel hesaplama.

Kod Açıklaması:

Fakt Fonksiyonu: Alınan değerin faktöriyelinin hesaplanması.

Calculate Fonksiyonu: Alınan prob(A takımının kazanma olasılığı) ve size(toplam oynanacak maç serisi) değerlerini kullanarak dinamik programlama ile A takımının bu seride kazanma olasılığını hesaplar. Fonksiyon dinamik programlama özelliklerine uygun olarak bir matris açar A takımı üzerinden işlem yapacağımız için matrisimizin sol tarafı A takımının olasılıklarını temsil eder ve satır sayımız seriyi kazanmak için gereken minimum kazanma sayısına eşittir, sütun sayımız ise bu

değerin bir eksiğine eşittir çünkü B takımının kazanma ihtimalini göz ardı ediyoruz.

- Bu yapıya uygun olarak matrisin ilk elemanı olasılık hesaplayacağımız için çarpım durumunda kendisini vermesi adına 1 değeri yazılır.
- Sütun veya satır değerlerinde karşı takımın olasılığından bağımsız olarak sadece belirli takım kazanacağından olasılık değerimizle çarpılarak ilk satır ve ilk sütun doldurulur.
- Diğer bütün durumlarda ilk önce satır başından A takımının kazandığı maç sayısının olasılığı*, sütun başından B takımının kazandığı maç sayısının olasılığı hesaplanır**, ardından iki takımın birbirlerine karşı hangi sırayla maç kazandığını bilmediğimizden ötürü bu kombinasyon da hesaplanıp çarpılır***. Ardından olasılık kümülatif bir işlem gerektirdiğinden bir önceki satırdaki değerle toplanarak ilerlenir.

* = Örneğin A takımının 2 maç kazanma olasılığı, oranı 0.4 düşünüldüğünde 0.16 olur.

** = Aynı örnekle devam edilirse bu durumda B takımının 1 maç kazanma olasılığı, oranı 0.6 düşünüldüğünde 0.6 olur.

*** = Bu şartlarda A takımının 2 maç B takımının 1 maç kazanmasının olasılığını hesaplıyoruz. Bu 3 maç A takımı açısından = Yapılan 2 veya 3 maçta

{kazan,kazan} + {kazan,kaybet,kazan},{kaybet,kazan,kazan}

Şeklinde 3 farklı durumda olabilir.

İlk önce üst üste 2 maç kazanma olasılığı bulunur ve 0. Sütunda yerini alır, ardından 3 maç içerisinde 2 maç kazanma olasılığı hesaplanır ve üst üste 2 maç kazandığı değerle toplanarak toplam olasılığımız hesaplanmış olur.

Yukarıdaki örnekten yola çıkarak, hesaplanacak ilk değer üst üste 2 maç kazanılma olasılığı 0.16 olarak bulundu. Ardından 3 maç içerisinde 2 maç kazanma olasılığı $(0.16) \times (0.6) \times (2)$ formülünden 0.192 olarak bulundu.

Son adımda ise bu 2 değer toplanarak 0.352 değeri bulundu.

Main Fonksiyonu:

A takımının kazanma olasılığı, Toplam oynanacak maç serisi alınır. Seri sayısı çift ise program sonlanır çünkü beraberlikle sonuçlanabilir. Ardından değerler calculate fonksiyonuna atılarak hesaplanır. Matris ve işlem sonucu ekranda gösterilir.

Ekran Görüntüleri:

- 1) Yukarıda Çözümümü açıklamak için verdiğim örneğin görüntülenmesi.

```
A takımının kazanma olasiligini giriniz: 0.4
Toplam mac sayisini giriniz: 3

Matris:
1.000000  0.600000
0.400000  0.640000
0.160000  0.352000

A takımının kazanma olasiligi: 0.352000
-----
Process exited after 2.769 seconds with return value 40
Press any key to continue . . .
```

2) Ödevde verilen örnekte bulunan 0.29 değerinin görüntülenmesi

```
A takiminin kazanma olasiligini giriniz: 0.4
Toplam mac sayisini giriniz: 7

Matris:
1.000000  0.600000  0.360000  0.216000
0.400000  0.640000  0.784000  0.870400
0.160000  0.352000  0.524800  0.663040
0.064000  0.179200  0.317440  0.455680
0.025600  0.087040  0.179200  0.289792

A takiminin kazanma olasiligi: 0.289792
-----
Process exited after 3.315 seconds with return value 40
Press any key to continue . . .
```

3) Ödevde sorulan 2. Sorunun gösterilmesi. (A takımının bir maçı kazanma olasılığının 0.6 olduğu durumda 7 maçlık seride (4 alan kazanır) kazanma olasılığı)

```
A takiminin kazanma olasiligini giriniz: 0.6
Toplam mac sayisini giriniz: 7

Matris:
1.000000  0.400000  0.160000  0.064000
0.600000  0.840000  0.936000  0.974400
0.360000  0.648000  0.820800  0.912960
0.216000  0.475200  0.682560  0.820800
0.129600  0.336960  0.544320  0.710208

A takiminin kazanma olasiligi: 0.710208
-----
Process exited after 8.142 seconds with return value 40
Press any key to continue . . .
```

C Kodu:

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

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

```
float fakt(int k){
```

```
    int i, f=1;          // Olasılık hesaplamak için gerekli faktoriyel  
fonksiyonu
```

```
    for(i=2;i<=k;i++)
```

```
        f = f*i;
```

```
    return f;
```

```
}
```

```
float calculate(float prob, int size){
```

```
    int length = size/2;
```

```
    float matrix[length+2][length+1];
```

```
    int i,j;
```

```
    for(i=0;i<length+2;i++){          // Kodun bu kısmının gerekli bütün  
açıklamaları detay ve örnekleriyle raporda açıklanmıştır.
```

```
        for(j=0;j<length+1;j++){
```

```
            if(i==0 && j==0)
```

```
                matrix[i][j] = 1;
```

```
            else if(j==0){
```

```
                matrix[i][j] = matrix[i-1][j] * (prob);
```

```
            }
```

```

        else if(i==0){
            matrix[i][j] = matrix[i][j-1] * (1-prob);
        }
        else{
            matrix[i][j] = (matrix[i][0]*matrix[0][j]) * ( fakt(i+j-
1) / ( fakt(i-1)*fakt(j) ) );
            matrix[i][j] += matrix[i][j-1];
        }
    }
}

```

```

printf("\nMatris:\n"); // Matrisin gösterilmesi

```

```

for(i=0;i<length+2;i++){
    for(j=0;j<length+1;j++){
        printf("%.6f ",matrix[i][j]);
    }
    printf("\n");
}

```

```

    return matrix[length+1][length]; // Dinamik programlama sonucu
oluşan ve aradığımız degerin son hücreden return edilmesi
}

```

```

int main(){
    float prob; // A takımının kazanma olasılığı
    int size; // Toplam mac sayısı ( serisi )
}

```

```
printf("A takiminin kazanma olasiligini giriniz: ");  
scanf("%f",&prob);
```

```
printf("Toplam mac sayisini giriniz: ");  
scanf("%d",&size);
```

```
if(size%2==0){ // Cift sayı girilmesi durumunda beraberlik  
olabileceginden program sonlanır.
```

```
    printf("Lutfen tek sayi giriniz");  
    return 0;
```

```
}
```

```
float result = calculate(prob,size); // Dinamik programlama sonucu  
elde edilen deger result degiskenine atılır ve ekranda gosterilir.
```

```
printf("\nA takiminin kazanma olasiligi: %f",result);
```

```
}
```