



# ALGORİTMA ANALİZİ

## ÖDEV 3 - GRUP 1

*İrem ATILGAN*

17061036



20.12.2020

# SORU 1

## 1.1 Çözüm

$$a) P(i,j) \begin{cases} 0, & i \neq 0 \text{ and } j = 0 \\ 1, & i = 0 \text{ and } j \neq 0 \\ P(i,j) = p * P(i-1,j) + (1-p) * P(i,j-1) & i \neq 0 \text{ and } j \neq 0 \end{cases}$$

b)

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.).

	1	1	1	1
0	0.6	0.84	0.936	0.9744
0	0.36	0.648	0.8208	0.91296
0	0.216	0.4752	0.68256	0.8208
0	0.1296	0.33696	0.54432	0.710208

Cevap = 0.710208 (%71.0208 olasılık)

## 1.2 Fonksiyonlar

**double\*\* createMatrix(int dim)** : Matris oluşturma fonksiyonu; parametre olarak kazanılması gereken maç sayısı alınır. Matris dönlür.

**double calcProb(double prob, int win)** : Dinamik programlama yolu ile olasılık hesaplayan fonksiyon; parametre olarak A takımının bir maçı kazanma olasılığı ve kazanılması gereken maç sayısı alınır. A takımının seriyi kazanma olasılığı sonuç olarak dönlür.

## 1.3 Program Kodu

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  //Matris olusturma fonksiyonu
5  double** createMatrix(int dim)
6  {
7      int i;
8      double **matrix;
9      matrix = calloc(dim, sizeof(double*));
10     for(i = 0; i < dim; i++)
11     {
12         matrix[i] = calloc(dim, sizeof(double));
13     }
14
15     return matrix;
16 }
```

```

18 //Dinamik programlama yolu ile olasilik hesaplayan fonksiyon
19 double calcProb(double prob, int win) {
20     int i,j;
21     double **matrix;
22     matrix = createMatrix(win+1); //Matris olusturulur
23     //A takiminin kazanmasi gereken mac kalmadiysa P(0,j) seriyi kazanma olasiligi 1'dir
24     for(i = 0; i <= win; i++)
25         matrix[0][i] = 1;
26
27     //B takiminin kazanacak maci kalmadiysa seriyi kazanmislardir
28     //Bu durumda A'nin seriyi kazanma olasiligi sifirdir
29     for(i = 0; i <= win; i++)
30         matrix[i][0] = 0;
31
32     //2. satir ve 2. sutundan itibaren (ilk satir yukarida dolduruldu) hucrelerin degeri hesaplanir
33     for(i = 1; i <= win; i++)
34     {
35         for(j = 1; j <= win; j++)
36         {
37             //P(i,j) = probA * P(i-1,j) + probB * P(i,j-1) recursive fonksiyonu kullanilir
38             matrix[i][j] = matrix[i-1][j]*(prob) + matrix[i][j-1]*(1-prob);
39         }
40     }
41     printf("\nMatrix\n-----\n");
42     for(i = 0; i <= win; i++)
43     {
44         for(j = 0; j <= win; j++)
45         {
46             printf("%lf\t",matrix[i][j]);
47         }
48         printf("\n");
49     }
50
51     return matrix[win][win];
52 }

56 int main()
57 {
58     int num_games; //Serideki toplam oyun sayisi
59     int win; //Seriye kazanmak icin kazanilmasi gereken oyun sayisi
60     double result; //Sonuc olasiligin tutuldugu degisken
61     double p; //A takiminin bir maci kazanma olasiligi
62     printf("Total # of match in series = ");
63     scanf("%d",&num_games);
64     printf("Possibility of winning one match for A team = ");
65     scanf("%lf",&p);
66     win = num_games/2 + 1;
67
68     result = calcProb(p,win);
69     printf("\nPossibility of winning series for A is %lf%%\n",result*100);
70
71     return 0;
72 }

```

## 1.4 Ekran Çıktısı

```
Total # of match in series = 7
Possibility of winning one match for A team = 0.6

Matrix
-----
0.000000      1.000000      1.000000      1.000000      1.000000
0.000000      0.600000      0.840000      0.936000      0.974400
0.000000      0.360000      0.648000      0.820800      0.912960
0.000000      0.216000      0.475200      0.682560      0.820800
0.000000      0.129600      0.336960      0.544320      0.710208

Possibility of winning series for A is 71.020800%
-----
Process exited after 5.311 seconds with return value 0
Press any key to continue . . .
```