



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM3021 Algoritma Analizi

Emre ÇELİK
15011902

2020-2021 Bahar Dönemi
Ödev 3 – Problem 1

Danışman: Doç. Dr. Mine Elif KARSLIGİL

İSTANBUL, 2020

Problem 1: 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'nın seriyi kazanma olasılığı $P(i,j)$ 'dir.

Programın kodları ve çalışmasını anlattığım video: <https://youtu.be/YDRa5u00vTQ>

Programımızda DEBUG mevcuttur. Kod başlangıcında DEBUG kısmını 'true' yaparak işlem adımlarını detaylıca inceleyebilirsiniz. Program macOS işletim sisteminde yazılmıştır.

a-) Problemin rekürans bağıntısı:

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

b-) Kazanma olasılığı 0.6 iken 7 maçlık seride (4 maçı alan kazanır) A takımın kazanma olasılığı:

```
15011902 — -zsh — 80x14
(base) emrecelik@MacBook-Air 15011902 % gcc -o 15011902_1 15011902_1.c
(base) emrecelik@MacBook-Air 15011902 % ./15011902_1
Recursive Approach: 0.710208
Dynamic Table
1.000000 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
Dynamic Approach: 0.710208
(base) emrecelik@MacBook-Air 15011902 %
```

c-) Problemin dinamik yaklaşımla çözüm kodu aşağıdadır.

```
// Dynamic programming approach
double series() {
    int i;
    int j;
    double matrix[n+1][n+1];
    clearMatrix(matrix);
    matrix[0][0] = 1;
    for (i = 1; i <= n; i++) // Team that we are looking for, because of that, we return 1
        matrix[0][i] = 1;

    for (i = 1; i <= n; i++) { // Calculating the win rate
        for (j = 1; j <= n; j++)
            matrix[i][j] = (WINRATE*matrix[i-1][j]) + ((1-WINRATE)*matrix[i][j-1]);
    }
    if(DEBUG)
        printMatrix(matrix);
    return matrix[n][n]; // returns the win rate
}
```

Örnekler:

Kazanma olasılığı 0.4 iken 7 maçlık seride (4 maçı alan kazanır) A takımın kazanma olasılığı:

```
15011902 — -zsh — 80x7
[(base) emrecelik@MacBook-Air 15011902 % gcc -o 15011902_1 15011902_1.c
[(base) emrecelik@MacBook-Air 15011902 % ./15011902_1
Recursive Approach: 0.289792
Dynamic Approach: 0.289792%
(base) emrecelik@MacBook-Air 15011902 %
```

Kazanma olasılığı 0.25 iken 7 maçlık seride (4 maçı alan kazanır) A takımın kazanma olasılığı:

```
15011902 — -zsh — 80x14
[(base) emrecelik@MacBook-Air 15011902 % gcc -o 15011902_1 15011902_1.c
[(base) emrecelik@MacBook-Air 15011902 % ./15011902_1
Recursive Approach: 0.070557
Dynamic Table
1.000000 1.000000 1.000000 1.000000 1.000000
0.000000 0.250000 0.437500 0.578125 0.683594
0.000000 0.062500 0.156250 0.261719 0.367188
0.000000 0.015625 0.050781 0.103516 0.169434
0.000000 0.003906 0.015625 0.037598 0.070557
Dynamic Approach: 0.070557%
(base) emrecelik@MacBook-Air 15011902 %
```