



**BİTLİS  
EREN  
ÜNİVERSİTESİ**

ADI: MERLİN DE L OR  
SOYADI: AKENMOE KAMCHE  
BÖLÜM: BİLGİSAYAR MÜHENDİSLİĞİ

#### AÇIKLAMA :

Çalışma zamanının kesin olarak belirlenmesi zordur

Giriş verilerine bağlı olan en iyi durum

Ortalama durum; hesaplanması zordur

Diğerlerine göre en kötü durum hesaplanması kolaydır

EN KÖTÜ ZAMAN VE EN İYİ ZAMAN

$j=n$ ,  $i=n$ ,  $k=n$  olsun

1. Dizideki sayıların toplamını bulma

	Birim zaman	Frekans	Toplam
Static int DiziToplami(int[] dizi){			
int toplam = 0;	1	1	1
for(int i = 0; i < dizi.Length; i++){	1,1,1	1,(n+1),2	2n+2
toplam += dizi[i];	1	n	n
return toplam;	1	1	1

$$T(N) = 3n+3$$

$$T \text{ Worst } (N) = O(n)$$

$$T \text{ Best } (N) = \Theta(n)$$

2. Matris Çarpım

	Birim zaman	Frekans	Toplam
static int[,] MatrisCarpim(int[,] matris1, int[,] matris2){			
int satir = matris1.GetLength(0);	1	1	1
int sutun = matris2.GetLength(1);	1	1	1
int[,] sonuc_matris = new int[satir,sutun];	1	1	1
for(int i = 0; i < satir; i++){	1,1,1	1,(n+1),n	2n+2
for(int j = 0; j < sutun; j++){	1,1,1	N(1,(n+1),n)	2n^2+2n
int toplam = 0;	1	n(n)	N^2
for (int k=0; k<matris2.GetLength(0);k++){	1,1,1	N(n(1,(n+1),n)	2n^3+2n^2
toplam += matris1[i, k] * matris2[k, j];}	1	N(n(n)	n^3
sonuc_matris[i, j] = toplam;}	1	N(n)	n^2
}return sonuc_matris;}	1	1	1

$$T(N) = 3n^3+6n^2+4n +6$$

$$T \text{ Worst } (N) = O(n^3)$$

$$T \text{ Best } (N) = \Theta(n^3)$$

### 3. Dizide bir elemanın aranması

	Birim Zaman	Frekans	Toplam
static int ikiliArama(int[] dizi, int sayi)			
int uzunluk = dizi.Length;	1	1	1
int sag = uzunluk - 1;	1	1	1
while (sol <= sag){	1	$\log_2 n + 1$	$\log_2 n + 1$
int ort = (sol + sag) / 2;	1	$\log_2 n$	$\log_2 n$
if (dizi[ort] == sayi){	1	$\log_2 n$	$\log_2 n$
return ort;}	1	1	1
else if (sayi < dizi[ort]){	1	$\log_2 n - 1$	$\log_2 n - 1$
sag = ort - 1;}	1	$\log_2 n - 1$	$\log_2 n - 1$
else{ sol = ort + 1; } }	1	$\log_2 n - 1$	$\log_2 n - 1$
return -1;}	1	1	1

$$T(N) = 6\log_2 n + 2$$

$$T \text{ Worst } (N) = O(\log_2 n)$$

	Birim zaman	Frekans	Toplam
static int ikiliArama(int[] dizi, int sayi)			
int uzunluk = dizi.Length;	1	1	1
int sag = uzunluk - 1;	1	1	1
while (sol <= sag){	1	$\log_2 n + 1$	$\log_2 n + 1$
int ort = (sol + sag) / 2;	1	$\log_2 n$	$\log_2 n$
if (dizi[ort] == sayi){	1	$\log_2 n$	$\log_2 n$
return ort;}	1	1	1
else if (sayi < dizi[ort]){			
sag = ort - 1;}			
else{ sol = ort + 1; } }			
return -1;}			

$$T(N) = 3\log_2 n + 4$$

$$T_{\text{Best}}(N) = 3\log_2 n + 4$$

$$T_{\text{Best}}(N) = \Theta(\log_2 n)$$