14. TEKRARLI BAĞINTILAR

Analizi yapılacak olan algoritma özyinelemeli veya iter aktif olabilirler. Bunun anlamı bir problemin çözümü birden fazla küçük problemin çözümlerinin bir araya getirilmesi ile olr. Bunun için tekrarlı bağıntılar kullanılabilir.Bu şekilde özyinelemeli bir programdan özyinelemeli fonksiyonlar grubuna bir eşleştirme yapılmış olur.

Bir tekrarlı bağıntı kanalı içerinde oldukça bilgi barındırdığından bir algoritmanın performansı analizinin yapılması için geliştirilen tekrarlı bağıntı önemli bir adımı teşkil eder.Giriş verilerine göre algoritmanın önemli özellikleri basit bir matematiksel ifade ile temsil edilmiş olurlar.Bir çok önemli algoritma özyinelemeli olarak tanımlanabilirler ve bu algoritmaların tekrarlı bağıntıları elde edilmiştir(Bu bağıntılar ortalama değer elemanı veya en kötü durum)

İlk olarak tekrarlı bağıntıların temel özelliklerini incelenebilmesi ve bu tekrarlı bağıntıların sınıflandırılması için yolların tespit edilmesidir.Ondan sonra birinci mertebeden tekrarlı bağıntı çözümleri ele alınabilir.Birinci mertebeden T.B parametresi için bir fonksiyonun parametresi n-1 olan fonksiyon türünden ifade edilmesidir.Daha sonra yüksek mertebeli ve sabit katsayılı lineer T.B çözümüne bakılacaktır. Ondan sonra nonlineer ve değişken tekrarlı bağıntıların yaklaşım çözümleri bakılacaktır.

Algoritma analizinde önemli bir yere sahip olan böl-yönet T.B önemli bir yere sahiptirler.Böl-yönet yöntemi ile geliştirilen merge sort algoritmasının.T.B bu sınıfa iyi bir örnektir.

T.B'in diğer bir dalı fark denklemidir.Çünkü T.B fark denklemleri şeklinde ifade edilirler ve $f_n=f_n-f_{n-1}$ olur. T.B 'in çözümleri diferansiyel denklem çözüm yöntemleri ile yapılırlar.

Bu bölümde algoritma analizinde ortaya çıkan T.B üzerinde durulacaktır ve bunların çözümleri yapılacaktır. Diferansiyel denklem çözüm yöntemleri kullanılacaktır. Bu yöntemler yerine ? fonk. kullanılabilmektir.

T.B Temel Özellikleri

Quick Sort ve mergesort algoritmalarının analizinde ortaya çıkan T.B

$C_N= (1+1/N)\ C_{N-1}+2$ ve $_{c1}=2$, N>1 (X.1)

$C_N= C_{\lfloor N/2 \rfloor}+ C_{\lceil N/2 \rceil}+N$ ve c1=0, N>1 (X.2)

$C_N= N+1+1/N \sum (C_{j-1} + C_{n-j})$, N>0 ve $C_o==$ (X.3)

$1 \le j \le N$

Merge Sort Algoritması Algoritma X.1'de görülmektedir.

Algoritma X.1 =Merge Sort (1,r ε z^+)
Yerel Değişkenler: i,j,k,m ε z
1.Eğer r-1>0 ise
2. m (r+l) /2
3. mergesort(l,m);
4. mergesort(m+1,r);
5. i 1,…,m-l+1
6. B[i] A[l+i-1]
7. j m+1,…,r
8. C[j-m] A[j]
9. i ,j , B[m+1] max
10. k 1 , ...r
11. Eğer B[i] < [j]
12. A[k] B[i]
13. j i+1
14. değilse
15. A[k] [j]
16. B j+1

Merge Sort algoritması,N elemanlı bir diziyi alır ve bu diziyi alt dizilere böler. Her alt dizi bir veya iki elemanlı olana kadar bölünür ve ondan sonra sırlama işlemi başlar.

Merge Sort herhangi bir sıralama algoritması kadar etkili olan bir algoritma olduğundan önemli bir algoritmadır. Sıralama algoritmalarında karşılaştırma sayısı önemlidir.

Teorem X.1: *Merge Sort algoritması ile n elemanlı bir dizi sıralanırken yapılan karşılaştırma sayısı NLgN + O(N)' dir.*

Not**:** Eğer L_n karşılaştırma sayısı ise ilk yarıyı sırlama için yapılan karşılaştırma sayısı $C_{\lfloor N/2 \rfloor}$ olur, sağ parçanın sıralaması için $C_{\lfloor N/2 \rfloor}$ tane karşılaştırma sayısı yapılır ve birleştirme için N tane karşılaştırma yapılır. Bunun sonucunda bir tekrarlı bağıntı

$C_N = C_{\lfloor N/2 \rfloor} + C_{\lfloor N/2 \rfloor} + N$ $c_1=0$ ve $N \ge 2$ (x.+) elde edilir. $N= 2^n$ için

$C_2^n = 2.C_2^{n-1} + 2^n$

Olur ve $n \ge 1$ $c_1=0$ denklemin her iki tarafı 2^n bölündüğünde

$C_2^n/2^n = C_2^{n-1}/_2^{n-1} + 1 = C_2^{n-2}/_2^{n-2} + 2 = C_2^{n-3}/_2^{n-3} + 3 = \ldots\ldots = C_2^0/_2^0 + n = n$ olur. Bu eşitliklerin doğruluğu

$C_2^{n-1}=2.C_2^{n-2}+ {}_2^{n-1} \Rightarrow C_2^n= 2.(2C_2^{n-2}/{}_2^{n-1}) + 2^n$
$= 2^2\ C_2^{n-2}+2.2^n$
$C_2^{n-2}=2.C_2^{n-3}+ {}_2^{n-2} \Rightarrow C_2^n= 2^2.(2C_2^{n-3}/{}_2^{n-2}) + 2.2^n$
$= 2^3\ C_2^{n-3}+3.2^n$
$C_2^{n-3}=2.C_2^{n-4}+ {}_2^{n-3} \Rightarrow C_2^n= 2^3.(2C_2^{n-4}/{}_2^{n-3}) + 3.2^n$
$= 2^4\ C_2^{n-4}+4.2^n$

şeklinde gösterilir.
$C_2^n=n.2^n \Rightarrow C_2^n \approx N.LgN$ olur.
$N \neq 2^n$ olduğun zaman analiz biraz daha karışıktır.
(X.2) denklemi $N=2^n$ için ...? ve $N \neq 2^n$ için tümevarım ile yapılır.(X.3) denklemi ise N-1 için ikinci denklem yazılır ve taraf tarafa çıkarma yapılır. Bunun sonucunda (X.1) elde edilir. Fakat bu yöntemler her lineer T.D uygulanamıyor. Bunlara örnek olarak
$F_n=F_{n-1}+F_{n-2}$, $F_0=0$, $f_1=1$;$n>1$
fibonacci serisi verilebilir
T.D sınıflandırılırsa aşağıdaki liste elde edilir
Birinci mertebe
Lineer $a_n=n.a_{n-1}-1$
Nonlineer $a_n=1/(1+ a_{n-1})$
İkinci mertebe
Lineer $a_n=a_{n-1}+2.a_{n-2}$
Nonlineer $a_n=a_{n-1}.a_{n-2} + \sqrt{a_{n-2}}$
Değişken katsayılı $a_n=n.a_{n-1} + (n-1)a_{n-2} + 1$
t. mertebe
$a_n=f(a_{n-1}, a_{n-2},....., a_{n-t})$
tüm tarihi $a_n=n+a_{n-1}+ a_{n-2} +....., a_1$
bül ve yönet $a_n=a_{\lfloor n-1 \rfloor}+ a_{\lceil n-2 \rceil}+n$
Tekrarlı bağıntılara çözümü küçük n değerleri için el ile yapılabilir fakat n büyüdüğü zaman bu işlem zahmetli bir işlem olmaya başlar . T.B özelliklerinden ölçekleme ve öteleme önemli özelliklerdir. Örneğin $a_n=f(a_{n-1})$, $n>0$ ve $a_0=1$
İçin $a_0=2$ yapılırsa dizinin bütün elemanları değişir ve bu işleme **ölçekleme** denir.

Lineerlik: Birden fazla başlangıç değeri olan lineer T.B. ölçeklenmesi başlangıç değerlerinin birbirinden bağımsız olarak değiştirilmesi ve bunun sonucunda elde edilir çözümlerin birleştirilmesi ile elde edilir. Eğer f(x,y) bir liner fonksiyon ve f(0,0)=0 ise,

$a_n=f(a_{n-1}, a_{n-2})$ n>1

çözümü

$u_n=f(u_{n-1}, u_{n-2})$ n>1 $u_0=1$, $u_1=0$

denklem çözümlerinin a_0 ile çarpılması sonucunun toplanması ile elde edilir.f(0,0)=0 şartı denklemi homejen yapar . Eğer f(.,.) fonksiyon sabit varsa, hesaba katılması zorunludur.

Birinci Mertebeden T.B

$a_n=x_n.a_{n-1}$, n>0 , $a_0=1$

denklemi

$$a_n=\prod_{1\le k\le n} X_k$$

şeklinde ifade edilebilir. Böylece eğer $X_n=n$ ise

$a_n=n!$ Olur ve eğer $X_n=2$ ise $a_n=2^n$ olur.

Bu bir basit iterasyon yöntemidir. Iterasyon yöntemi kendine uygulanır ve taki bir sabit ile başlangıç değeri kalana kadar devam edilir. Iterasyon

$a_n=a_n.+Y_n$, n>0 ve $a_0=0$

uygulandığında

$$a_n=\prod_{1\le k\le n} Y_k$$

elde edilir. Böylece eğer $Y_n=1$ ise $a_n=n$ olur.

Eğer $Y_n=n-1$ ise $a_n=n(n-1)/2$ $1\le k\le n$ olur.

Teorem: *$a_n= x_n.a_{n-1}+ Y_n$ (n> 0, $a_0=0$) T.B*

Çözümü

$$a_n=X_nX_{n-1} ,....X_1 \quad \sum_{1\le j\le n} Yj/(X_j.X_{j-1} ...x_1)$$

$$= Yn+ \sum_{1\le j\le n} Yj.X_{j+1}, X_{j+2} ...x_n$$

elde edilir. Her iki tarafa X_{n+1}. X_{n+2}..... sayısı ile çarpılıp iterasyon yapılırsa aynı sonuç elde edilir.

Bu teorem

$C_n=(1+1/N). C_{N-1}+2$, N>1 , $C_1=2$

Denkleminin çözümlenebileceğini iddia eder. Her iki taraf

$N+1/N \cdot N/N-1 \cdot N-1/N-2.....3/2.2/1 = N+1$

Sayısına bölünür. Çözüm $2(N+1)(H_{N+1})$ olarak elde edilir.

$C_N=N+1/N \cdot C_{N-1} + 2$

$N/N+1.C_N= C_{N-1} + 2$

$N.H_{N+1}.C_N= C_{N-1} + 2. H_{N+1}$

NONNİNEER BİRİNCİ MERTEBE T.B

Bir T.B'yle a_n ve a_{n-1} parametrelerine bağlı bir nonlinner fonksiyona bağlı ise bu durumda ..? durumlar ortaya çıkar. Bu durumda kapalı forum çözüm beklenmez.

$n>0$ ve $A_0=1$ için $a_n=1/(1+a_{n-1})$ denkleminin çözümünde başlangıç değerleri önemlidir. Çünkü genelde çözüm bir sabite yakınsar. Bu bağıntıda terim sayısının her artışında oran biraz daha belirginleşmeye başlar. (Değişmeyen digit sayısı artar.) Buna basit yakınsama denir. Eğer bu denklem bir sabite yakınsıyorsa bu sabittir.

$A=1/(1+\alpha)$

Eşitliğini sağlaması gerekir Bu denklem çözüldüğü zaman x=0.6180334 elde edilir.

QUADRATİK YAKINSAMA VE NEWTON METODU

Newton Yöntemi ile fonksiyonların köklerini bulmada kullanıla iteratif bir yöntemdir. Newton yöntemi ile şart sayısının karakökü hesaplanacak olursa, iterasyon yöntemi

$an= ½ (a_{n-1} + \beta/a_{n-1})$, $n>0$ ve $a_0=1$

bağıntısına uygulanır. $b_n=a_n-\alpha$ yapılıp a_n için yerine yazıldığı zaman

$b_n=b^2_{n-1}+\beta-\alpha^2/2(b_{n-1}+ \alpha)$

elde edilir. Eğer $\alpha=\sqrt{\beta}$ ise , yaklaşık olarak $b_n \approx b^2_{n-1}$ olur.

$a_n=a_{n-1}(1-a_{n-1})$ T.B düşünüldüğü zaman elde edilecek yakınsamaya **yavaş yakınsam** denir. Benzer T.B gelişi güzel ikili ağaçların yüksekliğinin analizinde kullanılır.

$\lim a_n=0$

$N \rightarrow \infty$

Olduğunun görülmesi zor değildir. Yakınsamanın hızını görmek için $1/a_n$ düşünülür ve

$1/a_n=1/a_{n-1} (1/1-a_{n-1})$

$=1/a_{n-1} (1 + a_{n-1} + a^2_{n-1} + ...)$

$>1/a_{n-1} + 1$

olur. Bu teleshop $1/a_n > n$ veya $a_n < 1/n$ sonucunu verir.

Sonuç olarak $a_n=O(1/n)$ olur. $a_n=f(a_{n-1})$ için üç tane durum ele alındı ve f fonksiyonu sürekli bir fonksiyondur. Eğer a_n α limitini yakınsıyorsa ve yerde şart olarak α'nın sabit nokta olması gerekiyor ve $\alpha=f(\alpha)$ olur.

İncelenen üç durum;

Eğer $0<|f'(\alpha)|<1$ ise yakınsama basittir.

Eğer $f'(\alpha)=0$ ise yakınsama kuadratiktir.

Eğer $|f'(\alpha)|=1$ ise yakınsama yavaştır.

Yüksek Mertebeli T.B

Bu bağıntılarda sağ taraf a_{n-1},a_{n-2},a_{n-3} parametrelerin lineer kombinasyonudur ve katsayıları sabittir. Örnek olarak

$a_n=3.a_{n-1}-2.a_{n-2}$, $n>1$, $a_0=0$,$a_1=1$

ilk olarak $.a_n-a_{n-1}=2(a_{n-1}-a_{n-2})$ özelliğinden bu bağıntı çözülebilir. Bu çarpım devam ederse

$a_n-a_{n-1}=2^{n-1}$

olur ve buradan $a_n=2^n-1$ olur. Bu denklem $a_n-2.a_{n-1}= a_{n-1}-2.a_{n-2}$ eşitliğinden de çözülebilir. Bu işlemler $1-3x+ 2X^2=(1-2x)(1-x)$ çarpımlarından gelmektedir.

Teorem: *$a_n=x_1.a_{n-1}+x_2.a_{n-2}$ +....+$x_t.a_{n-t}$, $n\geq t$*

eşitliğinin bütün çözümleri $n^j\beta^n$ şeklindeki terimlerin eşitliğinin bütün çözümleri $n^j\beta^n$ şeklindeki terimlerin lineer kombinasyonu olur.β değeri

$g(z)=z^t-x_1.z^{t-1}-x_2.z^{t-2}$..... x_t

karakteristik polinomunun bir köküdür ve eğer β'nın v çarpanı varsa $0\leq j\leq v$ olur.

İSPAT: $C_n=\beta^n$ şeklindeki çözümler aranır. Herhangi bir çözüm yerine yazıldığında

$$\beta^n=x_1.\beta^{n-1}+x_2.\beta^{n-2}+x_3.\beta^{n-3}+.....+x_t.\beta^{n-t} \quad n\geq t$$

eşitliğini sağlamalıdır veya eşitlik olacak.

$$\beta^{n-t}q(\beta)=n$$

olur. Yani β^n bir çözümdür. Eğer β değeri q(z) denkleminin katlı kökü olsun. Bu durumda β^n çözümünün yanında $n.\beta^n$ bir çözümdür. Yerine yazıldığında

$$n\beta^n= x_1(n-1)\beta^{n-1}+x_2(n-2)\beta^{n-2}+.....+x_t(n-t)\beta^{n-t} \text{ , } n\geq t$$

veya eşiti olarak

$\beta^{n-t}((n-t)q(\beta)+\beta q^1(\beta))=0$

olur. Eğer β katlı kök ise $q(\beta)=q^1(\beta)=0$ olur.

Sabit Olmayan Katsayılar

Katsayılar sabit değilse , daha ileri tekniklere ihtiyaç duyulur. Üreteç fonksiyonlar veya yaklaşım metoları bunun için kullanılabilir. Örneğin;

$a_n=n.a_{n-1}+ n(n-1).a_{n-2}+$, $n>1$,$a_1=1$ ve $a_0=0$ bağıntısı n! Sayısına bölünerek çözüm elde edilebilir.

T.B Çözümleri İçin Metotlar

Nonlineer veya değişken katsayılı T.B 'in çözümleri elde etmek normal olarak yapılabilir veya yaklaşım yöntemleri kullanılabilir. Dört tane genel metot vardır. Değişken değişimi , reperoire ,bootstrapping , perturbation

Değişken Değişimi

Eğer $bn=an/xn.xn-1....$ x1 şeklinde değişim yapıldığında iterasyon sonucunda toplam elde edilir. Örneğin $n>1$ ve $a_1=2$ için $an=\sqrt{}$ an-1.an-2 bağıntısı düşünülsün.

Her iki tarafın logaritması alındıktan sonra $b_n=\lg a_n$ değişken değişimi yapılır. Yani denklem

$b_n=1/2(b_{n-1}+ b_{n-2})$ $n>1$ $b_0= 0$ ve $b_1=1$ olur.(Sabit katsayılı lineer bağıntıdır.)

Diğer bir problemde

$a_n= a^2_{n-1} -2$

denklemidir.$a_0=2$ oldukça zaman $n>1$ için $a_n=2$ olur. $n>1$ ve $a_0=1$ için $a_n=-1$ olur. Genel çözüm elde etmek için $a_n=b_n+1/b_n$ değişken değişimi yapılır.

$$b_n+1/b_n=b^2_{n-1}+1/b^2_{n-1} \quad n>0 \quad b_0+1/b_0=a_0$$

denklemi oluşur. $b_n= b^2_{n-1}$ kabulü yapılarak iterasyon ile

$b_n= b_0{}^{2^n}$ olur.b_0 değeri a_0' dan hesaplanır ve b_0'nun son .....

$$b_0= \frac{1}{2} (a_0 \mp \sqrt{a_0{}^2 - 4} \text{ olur. Böylece}$$

$$a_n=(\frac{1}{2} (a_0+\sqrt{a_0{}^2 - 4}))^{2^n} + (\frac{1}{2} (a_0-\sqrt{a_0{}^2 - 4})^{2^n} \text{ olur.}$$

Reportoire: Bu yöntem lineer T.B uygulanır ve bu işlemin adımları

*Ekstra fonksiyon terimi ekleyerek T.B revaksiyon oluştur.

*Bilinen fonksiyonlar T.B eklemek T.B.'ya benzeyen terimler elde edilir.

*T.B aynı olan bir bağıntının lineer terimlerin kombinasyonu ile elde edilir.
Örneğin ; $a_n=(n-1).a_{n-1} - a_{n-2}$