BÖLÜM 15. ÖZDEVİNİM

Özdevinimli algoritmaların analizinin yapılabilmesi için üç farklı yöntem kullanılmaktadır. Bunlardan biri iterasyon yöntemidir ve bu yöntem iteratif algoritmalar içinde çalışır. Diğer bir yöntem yerine koyma yöntemidir ve bu yöntemde mertebenin alt ve üst sınırları için bir tahmin yapılır ve tümevarım kullanılarak yapılan tahminlerin doğru olup olmadığı test edilir. Diğer bir yöntem ise Master yöntemi olarak bilinir ve bazı algoritmalar için daha kolay karar vermeyi sağlar. Fakat bu yöntemin bazı boşlukları vardır ve bu boşluklar başka önerilerle giderilmeye çalışılmıştır. Buna rağmen iyi sonuç veren bir yöntemdir.

Bir algoritmanın Özdevinimli zaman bağıntısı (15.1)' de görülmektedir.

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ aT(n/b) + f(n) & n > 1 \end{cases} \qquad (15.1)$$

(15.1)' de verilen bağıntıda a=2 ve b=2 olarak alınır f(n)=Θ(n) olarak alınırsa, verilen algoritmanın mertebesi Θ(nlgn) olur. Algoritmaların kullandığı veri kümesinin boyutu tamsayı olacağından T(n) için verilen n parametresi her zaman tamsayıdır. Tamsayı olmayan durumlar için T(n) tanımlı değildir. Bilgisayar mühendisliğindeki önemli algoritmaların bir kısmı **böl-ve-yönet** mantığını kullanırlar (bu mantık sıralama algoritmaları kısmında açıklanacaktır). Bu mantık ile tasarlanan algoritmaların zaman bağıntısı tekrarlı bağıntı olur. Bundan dolayı tekrarlı bağıntıların asimptotik davranışlarının incelenmesi gerekir. n>0 için $a_n=3a_{n-1}$, bir Özdevinimli (tekrarlı) bağıntıdır. Tekrarlı bağıntılarda başlangıç şartı veya şartları verildiğinde bu bağıntının çözümünün yapılması adi diferansiyel denklemlerde olduğu gibidir. Başka bir çözüm yolu da iterasyon ile çözmedir. Örneğin $a_0=4$ olarak verildiğinde

$a_1=3.a_0=3.4=3.4$
$a_2=3.a_1=3.3.4=3^2.4$
$a3=3.a2=3.3^2.4=3^3.4$

....

$a_n=4.3^n$

olur. Buradan yola çıkarak k sabiti için tanımlanan $a_n=ka_{n-1}$, tekrarlı bağıntısının çözümü $an=k^n.a_0$ olur. Biraz önceki örnekte bunun böyle olduğu görüldü.

Eğer tekrarlı bağıntı

$$a_n=k_1a_{n-1}+k_2a_{n-2} \tag{15.2}$$

k_1 ve k_2 sabitleri için bu şekilde tanımlanırsa, iterasyon yöntemi iyi sonuç vermeyebilir. Bu durumda sabit katsayılı adi diferansiyel denklem çözme yöntemi kullanılır. $a_n=cr^n$ şeklinde bir çözüm olduğu kabul edilir ve bu çözüm bağıntıda yerine yazılır. Bu yolla elde edilecek çözüm homojen tekrarlı bağıntının çözümü olur. (15.2) bağıntısı için bu çözüm yerine konulursa

$$cr^n=k_1cr_{n-1}+k_2cr_{n-2}$$

olur. Her terim cr^{n-2} değerine bölünürse

$$r^2=k_1r+k_2 \tag{15.3}$$

şeklinde olan r' ye bağlı ikinci dereceden tek değişkenli bir denklem elde edilir ve bu denkleme tekrarlı bağıntının ***karakteristik denklemi*** denir. Bu denklemin çözümü yapıldığında r_1 ve r_2 gibi iki kök değeri elde edilir. Bu kök değerlerine ***karakteristik kök*** denir. Eğer $r_1 \neq r_2$ ise, bağıntının çözümü

$$a_n=c_1r_1^n+c_2r_2^n \tag{15.4}$$

olur. Başlangıç şartları kullanılarak c_1 ve c_2 sabitleri bulunur. Eğer $r_1=r_2$ ise, bağıntının çözümü

$$a_n=c_1r_1^n+c_2nr_1^n \tag{15.5}$$

olur.

Algoritma analizi yapılırken elde edilecek tekrarlı bağıntı algoritmanın kendi verisini kendisini çağırırken nasıl parametre olarak gönderdiği önemlidir. Bir veri kümesinin boyutu her zaman tamsayı olur. Veri kümesinin boyutu hiçbir zaman tamsayı olmayan bir gerçel sayı olamaz. Bundan dolayı özellikle böl-ve-yönet yönteminde veri kümesinin nasıl bölündüğü önemlidir. Taban ve tavan fonksiyonları kullanılır. Örneğin (15.1) bağıntısı Hızlı Sıralama algoritmasına göre

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \Theta(n), & n > 1 \end{cases} \qquad (15.6)$$

şeklini alır. Bunun anlamı ilk önce veri kümesinin $\lfloor n/2 \rfloor$ kadarı parametre olarak gönderilir ve bundan sonra da $\lceil n/2 \rceil$ kısmı parametre olarak gönderilir. $\Theta(n)$ anlamı ise bir sefer veri kümesini ikiye bölme ve algoritmanın kendisini bu parçalar ile tekrar çağırma zamanı ile aynı bölmenin çözümünün birleştirilmesi için harcanan zamanı gösteren bir fonksiyondur.

Yeterince küçük n değerleri için $T(n)=\Theta(1)$ olur. Bir algoritmanın analizinin yapılabilmesi için küçük n değerleri için sınır değerleri göz önüne alınmaz ve T(n) sabit kabul edilir. Büyük n değerleri için de sınır değerleri göz önüne alınmaz ve bu durumda (15.6)' te verilen bağıntı

$$T(n)=2T(n/2)+\Theta(n)$$

olur. Bundan dolayı algoritma analizinde tavan, taban ve sınır değerleri koşulları genellikle ihmal edilirler.

15.1. Yerine Koyma Yöntemi

Bazı tekrarlı bağıntıların çözümü yapılmadan çözümünün nasıl olabileceği hakkında tahmin yapılabilir. Yerine koyma yönteminde bu mantık kullanılmaktadır. Çözüm tahmini yapılır ve yerine konulur. Ondan sonra tümevarımla doğruluğunun ispatı yapılır. Bu yöntem ilk bakışta kolay ve uygulanabilir gibi görünüyor. Fakat yapısı biraz karışık olan tekrarlı bağıntılar için çözüm tahmini yapmak oldukça zordur. Bundan dolayı her tekrarlı bağıntının çözümü bu yolla yapılamaz.

Bir tekrarlı bağıntının alt ve üst sınır tahmini içinde bu yöntem kullanılabilir. Örneğin

$$T(n)=2T(n/2)+n$$

şeklinde verilen bir tekrarlı bağıntı için alt sınırın en kötü tahminle $\Omega(n)$ açıktır. Tecrübelere dayanarak üst sınırın ise $O(n\lg n)$ olduğu görülebilir. Bunun anlamı c gibi bir pozitif sabit için $T(n)\leq cn\lg n$ olur. Bu sınırın tavan ve taban fonksiyonları içinde sağlandığı kabul edilir ve

$$T(\lfloor n/2\rfloor)\leq c\lfloor n/2\rfloor\lg(\lfloor n/2\rfloor)$$

olur. Bu değerlerin tekrarlı bağıntıda yerine yazılmış şekli aşağıdaki gibidir.

$$\begin{aligned}T(n)&\leq c(\lfloor n/2\rfloor\lg(\lfloor n/2\rfloor))+n\\&\leq cn\lg(n/2)+n\\&=cn\lg n-cn\lg 2+n\\&=cn\lg n-cn+n\\&\leq cn\lg n\end{aligned}$$

$c\geq 1$ olduğu sürece son adım sağlanır. Yeterince büyük n değerleri için sınır değerlerinin sağlandığı gösterilebilir. Fakat küçük n değerleri için problem çıkabilir ve buna örnek olarak $T(1)\leq c1\lg 1=0$ durumu verilebilir. Fakat bir fonksiyonun asimptotik davranışının analizinde belli bir n_0 değerinden büyük n değerleri için bu durum incelenir. Bundan dolayı n_0 değeri belirlendikten sonra c sabiti bulunur. $T(1)=1$ olarak alınırsa $T(2)=4$, $T(3)=5$ olur ve

$$c2\lg 2\geq 4\Rightarrow c\geq 2$$

olur. Benzer şekilde $c3lg3 \geq 5$ için de bir c sabiti bulunabilir. Daha önce belirtildiği gibi her tekrarlı bağıntı için çözüm tahmini yapmak zordur. (15.5)' te verilen tekrarlı bağıntı için iyi bir çözüm tahmini yapmak oldukça zordur.

$$T(n)=a(T(n/b)+k)+f(n) \tag{15.7}$$

Bu konuda tecrübe sahibi olan biri ancak iyi bir çözüm tahmini yapabilir. n değeri çok büyük olduğu zaman T(n/b)+k değeri ile T(n/b) değeri arasındaki değer farkının asimtotik davranış analizinde fazla önemli olmayabilir. Bundan dolayı çok büyük n değerleri için

$$T(n)=a(T(n/b)+k)+f(n)$$

tekrarlı bağıntısı yerine

$$T(n)=aT(n/b)+f(n)$$

bağıntısı alınabilir ve bu bağıntı için çözüm tahmini yapılabilir. Başka bir yaklaşımda alt ve üst sınır için tahmin yapılmasıdır. Bu yaklaşımda yapılacak alt ve üst sınır tahminlerinin başlangıçta iyi olması beklenmez, doğru olması beklenir. Örneğin T(n) için alt sınır tahmini $\Omega(f(n))$ olarak alınması ve üst sınır için de $O(n^2)$ olarak alınması yanlış olmaz. Çünkü T(n) bağıntısının çözümünde $c_1 nlgn$ şeklinde baskın bir terim vardır (c_1 bir pozitif sabittir). Daha sonra alt ve üst sınırlar T(n) bağıntısının çözüm bölgesini sınırlayacak şekilde yaklaştırılmaya çalışılır.

Yapılabilen hatalardan biri de tümevarım kullanılırken yapılan hatalardır. Tümevarımın bütün adımları doğru olarak kullanılmaz veya adımların bir kısmı eksik bırakılır. Bunun sonucunda yapılan tahmin doğruymuş gibi sonuç elde edilebilir.

İlk bakışta bazı tekrarlı bağıntılar için tahmin yapılması zor görünebilir, fakat değişken değişimi yöntemi ile bilinen bir tekrarlı bağıntıya veya daha kolay bir yapıya sahip olan bir tekrarlı bağıntıya dönüşebilir. Örneğin (15.8)' de görülen bağıntı için değişken değişimi yapılabilir.

$$T(n)= 2T(\sqrt{n}) + \lg n \qquad (15.8)$$

Bu tekrarlı bağıntıda m=lgn olarak alınırsa, (15.9)' deki bağıntı elde edilir.

$$T(2^m) = 2T(2^{m/2}) + m \qquad (15.9)$$

Bu bağıntının çözümü için tahmin yapmak daha kolaydır. Bu bağıntı daha da sadeleştirilebilir ve $U(m)=T(2^m)$ olarak alınırsa

U(m)=2U(m/2)+m

bağıntısı elde edilir. Bu bağıntının üst sınırı için U(m)=O(mlgm) tahmini yapılabilir. T(n) bağıntısının üst sınırını bulmak için m değişkeninin yerine n değişkeni türünde eşiti yazılır.

U(m)=O(mlgm)=O(lgnlglgn)

olur. Bunun anlamı

T(n)=O(lgnlglgn)

üst sınır değeri vardır.

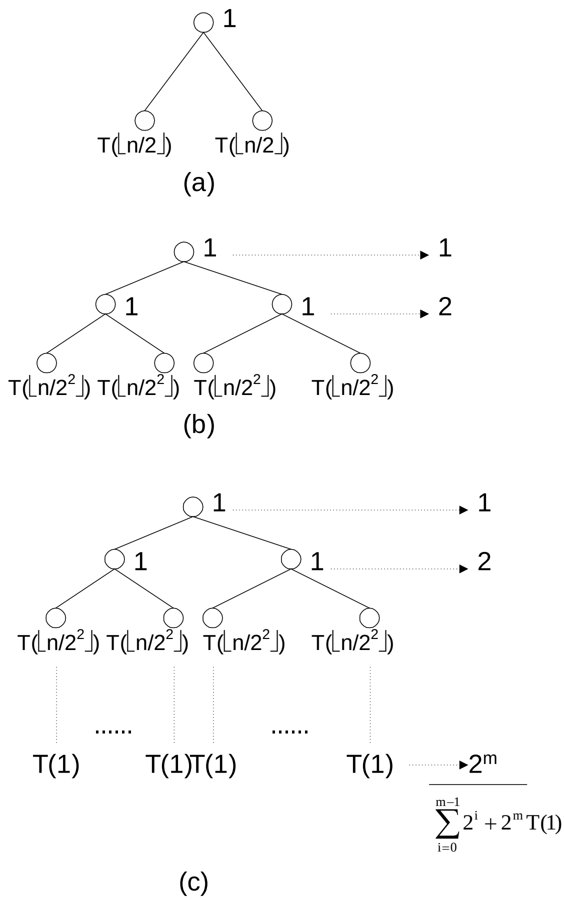
15.2. İteratif Yöntem

Bu yöntemde verilen bağıntının çözümünü bulmak için büyük endeksli terimin yerine küçük endeksli terim yazılarak, genel terimin çözümü için bir yargıya varılıncaya kadar bu işleme devam edilir veya başlangıç şartlarına kadar devam edilir. Örneğin

$T(n)=2T(\lfloor n/2 \rfloor)+1$

Bağıntısının çözümü için iteratif yöntem uygulansın.

$$
\begin{aligned}
T(n)=2T(\lfloor n/2 \rfloor)+1 \\
&=1+2(1+2T(\lfloor n/2^2 \rfloor)) \\
&=1+2(1+2(1+2T(\lfloor n/2^3 \rfloor))) \\
&=1+2+2^2+2^3T(\lfloor n/2^3 \rfloor) \\
&\ldots\ldots\ldots\ldots\ldots \\
&=\sum_{i=0}^{m-1} 2^i + 2^m T\left(\left\lfloor \frac{n}{2^m} \right\rfloor\right)
\end{aligned}
$$



Şekil 15.1. *T(n)=2T(⌊n/2⌋)+1 bağıntısı için Özdevinim ağaçları. (a) T(n) bağıntısının eşiti, (b) T(n) ağacının bir adım ilerletilmiş şekli, (c) T(n) bağıntısının genel çözümü.*

T(n) bağıntısının asimptotik davranışı için sınır belirlemek için son adımda elde edilen iki terimin sonucu bulunur. $n=2^m$ olduğunu kabul ederek ve T(1)=c gibi bir sabit olsun.

$$
\begin{aligned}
T(n) &= \frac{1-2^m}{1-2} + c2^m \\
&= \Theta(2^m) \\
&= \Theta(n)
\end{aligned}
$$

elde edilir. n değişkenin değeri 2' nin kuvveti ise, tavan ve taban fonksiyonun bu bağıntı için ihmal edilmesi herhangi bir hataya sebep olmaz. Bu yöntemin çözüm adımları ağaç şeklinde gösterilebilir ve elde edilen ağaca ***Özdevinim ağacı*** denir. Şekil 15.1' de çözümü yapılan örneğin Özdevinim ağacı görülmektedir.

Bu örnek için Özdevinim ağacının yüksekliği lgn olur. (15.1)' de verilen bağıntıda eğer n sayısının değeri b' nin kuvveti ise, Özdevinim ağacının yüksekliği $\log_b n$ olur. Eğer n değişkenin değeri b' nin kuvveti değil de genel bir tamsayı ise, Özdevinim ağacının yüksekliği $\lceil \log_b n \rceil$ olur. Şekil 15.1' deki Özdevinim ağacının yüksekliği kullanılarak T(n) bağıntısının hesabı yapılırsa,

$$
T(n)=\sum_{i=0}^{\lg n} 2^i = \frac{1-2^{\lg n+1}}{1-2} = 2n-1
$$

olur. Özdevinim ağacı ile verilen tekrarlı bağıntı ve zaman bağıntısı bu bağıntı olan algoritmanın çalışması görselleştirilmiş olur. Bu yolla algoritmanın mertebesi hakkında daha kolay bir yargıya varılır.

15.3. Master Yöntemi

Bu yöntem

T(n)=aT(n/b)+f(n)

şeklinde verilen bir bağıntının çözümünü yapmadan asimptotik davranışı hakkında bir yargıya varmayı yapmaya çalışır. Bu bağıntıya bakıldığında, algoritmanın mertebesinin alt sınırının en kötü ihtimalle f(n) fonksiyonun bir sabit ile çarpılmasıdır. Bu bağıntıda a≥1, b>1 ve f(n) asimptotik pozitif bir fonksiyondur. Bu yöntemin uygulanması için üç durum göz önüne alınır. Bu durumları incelemeden önce

$$T(n)=aT(n/b)+f(n)$$

şeklinde verilen bir bağıntının anlamı, zaman bağıntısı bu olan bir algoritma, her adımda gelen veri kümesini a tane parçaya bölmektedir ve her parçanın boyutu n/b olur. Algoritmanın veri kümesini bir sefer bölmesi ve daha sonra çözümlerin bir sefer birleştirilmesi zamanı ise f(n) olur. Asimptotik davranışın incelenmesinde $\lfloor n/b \rfloor$ veya $\lceil n/b \rceil$ yerine n/b yazılması hataya sebep olmaz. Bundan dolayı böl-ve-yönet algoritmalarının analizinde eğer taban veya tavan fonksiyonları kullanılmışsa, tekrarlı bağıntıda bu fonksiyonlar ihmal edildikten sonra analiz yapılır.

Master Teoremi

a≥1, b>1 ve f(n) bir fonksiyon olmak üzere T(n) bağıntısı

$$T(n)=aT(n/b)+f(n)$$

şeklinde verilip, n/b oranı $\lfloor n/b \rfloor$ veya $\lceil n/b \rceil$ anlamındadır. Bunlara bağlı olarak T(n) bağıntısının asimptotik davranışı için aşağıdaki yargılar söylenebilir.

a) Eğer ε>0 bir sabit için $f(n)=O\left(n^{\log_b a-\varepsilon}\right)$ ise, $T(n)=\Theta\left(n^{\log_b a}\right)$ olur.

b) Eğer $f(n)=\Theta\left(n^{\log_b a}\right)$ ise, $T(n)=\Theta\left(n^{\log_b a}\lg n\right)$ olur.

c) Eğer ε>0 sabiti için $f(n)=\Omega\left(n^{\log_b a+\varepsilon}\right)$ ve bütün yeterince büyük n değerleri ve c>1 sabiti için af(n/b)≤cf(n) ise, T(n)=Θ(f(n)) olur.

Bu teoremin temel mantığı f(n) ve $n^{\log_b a}$ fonksiyonlarının karşılaştırılmasıdır. Hangi fonksiyon büyükse, T(n) bağıntısının asimptotik davranışında o fonksiyon etkili olur. Eğer f(n)> $n^{\log_b a}$ ise, T(n)= Θ(f(n)) olur. Eğer f(n)< $n^{\log_b a}$ ise T(n)= $\Theta\left(n^{\log_b a}\right)$ olur. Eğer f(n)= $n^{\log_b a}$ ise, T(n)= $\Theta\left(n^{\log_b a} \lg n\right)$ olur.

Bu teoremin yetersiz olduğu durumlar var. Verilen tekrarlı bağıntı bu durumlardan birine uyuyorsa, master teoremi kullanılarak bu bağıntının asimptotik davranışı hakkında bilgi sahibi olunamaz.

Birinci durumda f(n) fonksiyonun $n^{\log_b a}$ fonksiyonundan küçük olması yetmez, aynı zamanda polinom olarak küçük olması gerekir. Birinci durumda f(n) fonksiyonu n^{ε} çarpılması sonucunda elde edilen fonksiyon ancak $n^{\log_b a}$ fonksiyonun Θ-notasyonu kümesinin bir elemanı olur. Son seçenekte ise f(n) fonksiyonun $n^{\log_b a}$ fonksiyonundan büyük olması yetmiyor, aynı zamanda bir düzenlilik şartı da eklenmektedir. Bu düzenlilik şartı ise, ε>0 ve c>1 sabitleri için af(n/b)≤cf(n) olmalıdır.

Master teoreminin yetersiz kaldığı durumlar şunlardır.

- f(n)< $n^{\log_b a}$, fakat polinom olarak küçük değilse, master teoremi kullanılarak bir sonuca varılamaz.

- f(n)> $n^{\log_b a}$, fakat düzenlilik şartı sağlanmıyorsa, bu durumda master teoremi kullanılarak T(n) bağıntısının asimptotik davranışı için bir yargıya varılamaz.

- f(n)> $n^{\log_b a}$ olabilir, fakat f(n) fonksiyonu polinom olarak $n^{\log_b a}$ fonksiyonundan büyük değilse, master teoremin bu durumda da yetersiz kalır. T(n) bağıntısının asimptotik davranışı için bir yargıya varılamaz.

Bazı bağıntılar üzerinde master teoreminin nasıl çalıştığına bir göz atılabilir.

1- T(n)=4T(n/2)+n şeklinde verilen bir tekrarlı bağıntı olsun. Bu durumda f(n)=n olur ve $n^{\log_b a}$=n^2 olur. nf(n)=n^2 olduğundan ε=1 olur.

f(n) fonksiyonu $n^{\log_b a}$ fonksiyonundan polinom olarak küçük olduğundan dolayı, master teoremi uygulanır ve T(n) bağıntısının asimptotik davranışı

$$T(n)= \Theta\left(n^{\log_b a}\right)$$

olur.

2- $T(n)=2T(n/2)+n^2$ olarak verilen bir tekrarlı bağıntı olsun. Bu durumda $f(n)=n^2$ olur ve $n^{\log_b a}=n$ olur. $f(n)=n$. $n^{\log_b a}=n^2$ olur. $\varepsilon=1$ ve $af(n/b)\leq cf(n) \Rightarrow 2f(n/2)\leq cn^2 \Rightarrow 2n^2/2\leq cn^2 \Rightarrow c>1$ gibi bir sabit bulur. f(n) fonksiyonu hem $n^{\log_b a}$ fonksiyonundan polinom olarak büyüktür ve hem de düzenlilik şartı sağlandığından dolayı $T(n)=\Theta(f(n))= \Theta(n^2)$ olur.

3- $T(n)=2T(n/2)+n$ olarak bir tekrarlı bağıntı verildiğinde, $f(n)=n$ olur ve $n^{\log_b a}=n$ olur. Bundan dolayı $T(n)=\Theta(f(n)\lg n) =\Theta(n\lg n)$ olur.

4- $T(n)=5T(n/7)+n\lg n$ olarak tanımlanmış bir tekrarlı bağıntı olsun. Bu durumda $f(n)=n\lg n$ olur ve $n^{\log_b a} = n^{\log_7 5}$ olur. $n^{\log_7 5}<n\lg n$ olur, fakat f(n) fonksiyonu polinom olarak $n^{\log_b a}$ fonksiyonundan büyük olmadığından master teoremi çalışmaz.

Master yönteminin çalışması n değişkenin değerlerinin iki gruba ayrılması ile incelebilir. Bu durumlar

a) $n=b^k$ ve $n\in Z^+\cup\{0\}$
b) $n\in Z^+$

şeklinde verilebilir.

Birinci durum için b bir pozitif tamsayı olmak üzere (b>1) n=1, b, $b^2,\ldots,b^i,\ldots$ şeklindeki değerlerini alır. Bu durumda tavan ve taban fonksiyonlarının tekrarlı bağıntının parametresine hiçbir etkisi olmaz. Bundan dolayı ilk olarak bu durumun incelenmesi alınmıştır ve bu değerler için master yönteminin nasıl çalıştığı anlaşılırsa, diğer bütün pozitif tamsayılar için master yönteminin nasıl çalıştığı daha rahat anlaşılır.

T(n)=aT(n/b)+f(n) şeklinde tanımlanan bir tekrarlı bağıntı olsun ve b>1. Buradan üç durum göz önüne alınarak inceleme yapılabilir.

Durum 1. a≥1, b>1 ve f(n) fonksiyonu b sayısının tamsayı kuvvetleri üzerinde tanımlanan negatif olmayan bir fonksiyon olsun. T(n) bağıntısı b sayısının tamsayı kuvvetleri üzerinde

$$T(n) = \begin{cases} \Theta(1) & \text{aksihalde} \\ aT(n/b) + f(n) & n = b^k \end{cases}$$

şeklinde tanımlansın (k pozitif bir tamsayıdır). T(n) bağıntısının asimptotik davranışı

$$T(n) = \Theta(n^{\log_b a}) + \sum_{k=0}^{\log_b a - 1} a^k f(n/b^k) \qquad (15.10)$$

olur. Bunun böyle olduğu Özdevinim ağacı ile veya T(n) tekrarlı bağıntısı iteratif olarak çözülmeye çalışılarak gösterilebilir. İteratif yolla T(n) bağıntısı

$$\begin{aligned} T(n) &= f(n)+aT(n/b) \\ &= f(n)+af(n/b+a^2T(n/b^2)) \\ &= f(n)+af(n/b)+a^2 f(n/b^2)+a^3T(n/b^3) \\ &\ldots\ldots\ldots\ldots\ldots \end{aligned}$$

$$= f(n)+af(n/b)+a^2 f(n/b^2)+\ldots..+a^{\log_b n-1} f(n/b^{\log_b n-1}) + a^{\log_b n} T(1)$$

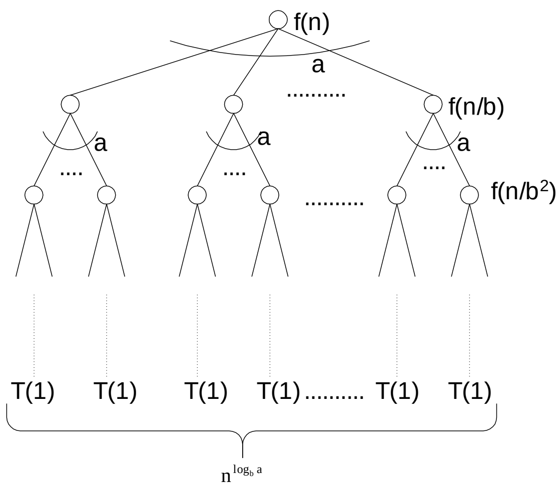
olur. Logaritmanın özelliklerinden $a^{\log_b n} = n^{\log_b a}$ olacağından yukarıdaki açılımın son terimi

$$a^{\log_b n} T(1) = \Theta(n^{\log_b a})$$

olur ve başlangıç şartları kullanıldığında T(1)=Θ(1) olur. Geriye kalan terimler ise

$$\sum_{k=0}^{\log_b n-1} a^k f(n/b^k)$$

olur. Böylece (15.10)' da verilen eşitliğin doğru olduğu gösterilmiş olur. Aynı sonucun Özdevinim ağacı kullanılarak elde edilebileceği de gösterilebilir ve T(n) bağıntısının Özdevinim ağacı Şekil 15.2' de görülmektedir.



Şekil 15.2. *T(n)=aT(n/b)+f(n) tekrarlı bağıntısının Özdevinim ağacı.*

Şekil 15.2' de görülen Özdevinim ağacının her seviyesindeki düğüm sayısı ile o seviyede en sağda gösterilen değer çarpımı o adımdaki işlem zamanın toplamını verir. Birinci seviyede f(n) olur. İkinci seviye için af(n/b) olurken, üçüncü seviye için $a^2f(n/b^2)$ olur. k. seviye için ise $a^{k-1}f(n/b^{k-1})$ olur. Özdevinim ağacının yüksekliği $\log_b n$ olduğuna göre son seviye için zaman bağıntısı $a^{\log_b n}T(1)$ olur. Bunun anlamı Özdevinim ağacının son seviyesinde $n^{\log_b a}$ tane yaprak vardır. T(1)=1 olarak alınırsa, zaman bağıntısının son seviyedeki karşılığı $n^{\log_b a}$ olur. Son seviye hariç diğer seviyelerdeki değerlerin toplamı da

$$\sum_{k=0}^{\log_b n-1} a^k f(n/b^k)$$

olur. Bu iki değerin toplamı yapılan iddiaya eşittir.

Durum 2. a≥1, b>1 ve f(n) fonksiyonu b sayısının tamsayı kuvvetleri üzerinde tanımlanan negatif olmayan bir fonksiyon olsun. g(n) fonksiyonu b sayısının tamsayı kuvvetleri üzerinde

$$g(n)=\sum_{k=0}^{\log_b n-1} a^k f(n/b^k)$$

şeklinde tanımlansın. g(n) fonksiyonu asimptotik olarak aşağıda hangi şartlarda hangi sınır değerlerini alacağı gösterilmiştir.

1- Eğer bir ε>0 sabiti için $f(n)=O\left(n^{\log_b a-\varepsilon}\right)$ ise, $g(n)=O\left(n^{\log_b a}\right)$ olur.

2- Eğer $f(n)=\Theta\left(n^{\log_b a}\right)$ ise, $g(n)=\Theta\left(n^{\log_b a}\lg n\right)$ olur.

3- Eğer bütün n≥b ve c<1 sabitleri için af(n/b)≤cf(n) ise, g(n)=Θ(f(n)) olur.

Bu üç durumun doğruluğu gösterilebilir. Eğer $f(n)=O\left(n^{\log_b a-\varepsilon}\right)$ ise,

$$f(n/b^k)=O\left((n/b^k)^{\log_b a-\varepsilon}\right)$$

olur. Bu değer g(n) fonksiyonunda yerine yazılırsa,

$$g(n)=O\left(\sum_{k=0}^{\log_b n-1} a^k\left(\frac{n}{b^k}\right)^{\log_b a-\varepsilon}\right)$$

elde edilir. Toplam sembolünün içinde indise bağlı olmayanlar toplam sembolünün dışına alınırlar.

$$\sum_{k=0}^{\log_b n-1} a^k\left(\frac{n}{b^k}\right)^{\log_b a-\varepsilon} = n^{\log_b a-\varepsilon}\sum_{k=0}^{\log_b n-1}\left(\frac{ab^{\varepsilon}}{b^{\log_b a}}\right)^k$$
$$= n^{\log_b a-\varepsilon}\sum_{k=0}^{\log_b n-1}\left(b^{\varepsilon}\right)^k$$
$$= n^{\log_b a-\varepsilon}\left(\frac{b^{\varepsilon\log_b n}-1}{b^{\varepsilon}-1}\right)$$
$$= n^{\log_b a-\varepsilon}\left(\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1}\right)$$

elde edilir. b ve ε sabit değerler olduğuna göre, parantez içindeki terimin karşılığı $O(n^{\varepsilon})$ olur. Buradan

$$g(n)= n^{\log_b a-\varepsilon}\, O(n^{\varepsilon})$$
$$=O(n^{\log_b a})$$

elde edilir. Benzer işlem ikinci durum içinde yapılabilir. İkinci durumda $f(n)=\Theta\left(n^{\log_b a}\right)$ olduğuna göre,

$$f(n/b^k)=\Theta\left((n/b^k)^{\log_b a}\right)$$

olur. Bu değerler g(n) fonksiyonunda yerine yazılırlarsa,

$$g(n)=\Theta\left(\sum_{k=0}^{\log_b n-1} a^k\left(\frac{n}{b^k}\right)^{\log_b a}\right)$$

elde edilir. Birinci durumda olduğu gibi toplam sembolünün indisinden bağımsız olan büyüklükler toplam sembolünün dışına alınabilirler.

$$\sum_{k=0}^{\log_b a-1} a^k\left(\frac{n}{b^k}\right)^{\log_b a} = n^{\log_b a}\sum_{k=0}^{\log_b n-1}\left(\frac{a}{b^{\log_b a}}\right)^k$$

$$= n^{\log_b a} \sum_{k=0}^{\log_b n-1} 1$$
$$= n^{\log_b a} \log_b n$$

elde edilir. Bu değer g(n) fonksiyonunda yerine yazılırsa,

$$g(n) = \Theta\left(n^{\log_b a} \log_b n\right)$$
$$= \Theta\left(n^{\log_b a} \lg n\right)$$

elde edilir ve ikinci durumun doğru olduğu gösterilmiş oldu. Son durumun ispatı için ise $c<1$ gibi bir sabit ve bütün $n \geq b$ değerleri için $af(n/b) \leq cf(n)$ olduğuna göre $a^k f(n/b^k) \leq c^k f(n)$ olur. Bu değerler g(n) için yerlerine yazılırlarsa,

$$g(n) \leq \sum_{k=0}^{\log_b n-1} a^k f(n/b^k)$$
$$\leq \sum_{k=0}^{\log_b n-1} c^k f(n)$$
$$\leq f(n) \sum_{k=0}^{\infty} c^k$$
$$= f(n)\left(\frac{1}{1-c}\right)$$
$$= O(f(n))$$

sonucu elde edilir. Bu şekilde don durumunda doğru olduğu gösterilmiş oldu.

Durum 3. $a \geq 1$, $b>1$ ve f(n) fonksiyonu b sayısının tamsayı kuvvetleri üzerinde tanımlanan negatif olmayan bir fonksiyon olsun. T(n) bağıntısı b sayısının tamsayı kuvvetleri üzerinde

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ aT(n/b) + f(n) & n = b^k \end{cases}$$

şeklinde tanımlansın (k pozitif bir tamsayıdır). T(n) bağıntısının asimptotik olarak b sabitinin kuvvetleri için şu şekilde sınırlanabilir.

1- Eğer bir $\varepsilon>0$ sabiti için $f(n)=O\left(n^{\log_b a-\varepsilon}\right)$ ise, $T(n)=\Theta\left(n^{\log_b a}\right)$ olur.

2- Eğer $f(n)=\Theta\left(n^{\log_b a}\right)$ ise, $T(n)=\Theta\left(n^{\log_b a}\lg n\right)$ olur.

3- Eğer $\varepsilon>0$ ve $c<1$ sabitleri için ve yeterli büyüklükteki bütün n değerleri için $af(n/b)\leq cf(n)$ ise, $T(n)=\Theta(f(n))$ olur.

Bu üç durumunda doğruluğu gösterilebilir. g(n) için bulunan sınır değeri kullanılarak

$$
\begin{aligned}
T(n) &= \Theta\left(n^{\log_b a}\right)+O(n^{\log_b a}) \\
&= \Theta\left(n^{\log_b a}\right)
\end{aligned}
$$

olur. Bu şekilde birinci durumun doğruluğu gösterilmiş oldu. İkinci durumun ispatı için ise,

$$
\begin{aligned}
T(n) &= \Theta\left(n^{\log_b a}\right)+\Theta\left(n^{\log_b a}\lg n\right) \\
&= \Theta\left(n^{\log_b a}\lg n\right)
\end{aligned}
$$

olur. Bu da ikinci seçeneğin doğru olduğunu gösterir. Üçüncü durumun ispatı için ise,

$$
\begin{aligned}
T(n) &= \Theta\left(n^{\log_b a}\right)+\Theta(f(n)) \\
&= \Theta(f(n))
\end{aligned}
$$

olur ve sonuncu durumun da doğru olduğu gösterilmiş oldu. n değişkeninin değerinin b sabitinin tamsayı kuvveti olmadığı durumlarda arzu edenler birkaç örnek üzerinde tavan ve taban fonksiyonlarını kullanarak nasıl sonuç vereceğine göz atabilirler. Burada anlatılanlar anlaşılmışsa, geriye kalan kısmın anlaşılması daha kolay olur ve bu kısım okuyucuya bırakılmıştır.

Bölüm Soruları

15.1. Asimptotik notasyonlardan hangilerinin geçişme özelliği vardır.

15.2. Aşağıdaki tekrarlı bağıntıları karakteristik denklem ve üreten fonksiyon yöntemleri ile çözünüz.

a) $a_n=5a_{n-1}-6a_{n-2}$, $a_1=36$ ve $a_0=0$
b) $a_n=3a_{n-1}-2a_{n-2}+2^{n-1}+2.3^n$, $a_1=29$ ve $a_0=9$
c) $a_n=a_{n-2}+4n$, $a_1=4$ ve $a_0=1$
d) $a_n=3a_{n-1}-2a_{n-2}+3.2^{2n-1}$, $a_1=12$ ve $a_0=0$

15.3. Aşağıdaki tekrarlı bağıntıların asimptotik davranışlarını inceleyiniz.

a) $a_n=2a_{n-1}-a_{n-2}+\Theta(n)$
b) $a_n=3a_{n-1}-2a_{n-2}+\Theta(n2^n)$
c)$a_n=4a_{n/2}-4a_{n/4}+\Theta(n\lg n)$
d) $a_n=5a_{n/2}-6a_{n/4}+\Theta(n)$

15.4. $T(n)=T(\lceil n/2\rceil)+1$ tekrarlı bağıntısı için $T(n)=O(\lg n)$ olduğunu gösteriniz.

15.5. $T(n)=2T(\lfloor n/2\rfloor)+n$ tekrarlı bağıntısı için $T(n)=\Omega(n\lg n)$ ve $T(n)=O(n\lg n)$ olduğunu gösteriniz. Son olarak $T(n)=\Theta(n\lg n)$ asimptotik davranışı gösterip göstermediğini açıklayınız.

15.6. Aşağıdaki tekrarlı bağıntıyı çözünüz.

$T(n)=T(pn)+T(qn)+\Theta(n)$
$T(1)=\Theta(1)$

15.7. i değişkeni 1 ile n-1 arasında değer alan düzenli dağıtık bir gelişigüzel değişken olsun.

$f(n)=(i+1)f(i)$ ve $f(1)=1$

tekrarlı bağıntısı için f(n) mümkün olan değerleri nelerdir ve ortalama değeri nedir?

15.8. $T(n) = 2T(\lfloor n/2 \rfloor + 25) + n$ tekrarlı bağıntısının çözümünün $O(n\lg n)$ olduğunu gösteriniz.

15.9. $T(n)=3T(\sqrt{2n})+2$ tekrarlı bağıntısını çözünüz.

15.10. $T(n)=3T(\lfloor n/5 \rfloor)+n$ tekrarlı bağıntısının çözümünü iteratif yolla gerçekleştiriniz. Bu bağıntının Özdevinim ağacı nedir?

15.11. Özdevinim ağacını kullanarak

$T(n)=T(n/3)+T(2n/3)+n$

bağıntısının çözümünü elde ediniz.

15.12. $b\geq 1$ bir sabit olmak üzere

$T(n)=T(n/b)+T(b)+n$

tekrarlı bağıntısının Özdevinim ağacını elde ediniz ve bu bağıntının çözümü nedir?

15.13. $0<a<1$ sabit olmak üzere

$T(n)=T(an)+T((1-a)n)+n$

tekrarlı bağıntısının Özdevinim ağacını elde ediniz ve asimptotik davranışı hakkında bilgi veriniz.

15.14. Master yöntemindeki düzenlilik şartının fiziksel anlamı nedir?

15.15. Master yöntemini kullanarak aşağıdaki tekrarlı bağıntıları çözünüz.

a) $T(n)=3T(n/3)+n$
b) $T(n)=3T(n/3)+n^2$

c) $T(n)=3T(n/3)+n^3$
d) $T(n)=3T(n/3)+n^k$

15.16. Aşağıdaki tekrarlı bağıntı verilmiş olsun.

$$T(n)=2T(\frac{n}{3})+\lg(n)$$

a) İteratif yöntem ile bu bağıntının mertebesini elde ediniz.
b) Master yöntemi ile bu bağıntının mertebesini elde ediniz.

15.17. Master teoremini kullanarak aşağıdaki bağıntının mertebesini elde ediniz.

$$T(n)=16T(n/4)+O(n^2)$$

15.18. $f(n)=n^2+4n\log n+900$ ve $g(n)=n^2+45\log n+h(n)$ fonksiyonları verilmiştir ve h(n) lineer olan bir polinomdur. f(n) ile g(n) arasındaki asimptotik ilişki nedir? f(n) ve g(n) arasındaki asimptotik ilişkiyi belirlerken h(n) polinomuna ihtiyaç var mıdır? Hangi durumlarda ihtiyaç duyulur veya duyulmaz?

15.19. Aşağıdaki tekrarlı bağıntıyı çözünüz.

$C_0=0$
$C_n=n+1+(2/n)(C_0+C_1+...+C_{n-1})$