

Algoritma Analizi

Ders 5: Böl ve Yönet Yaklaşımı

Doç. Dr. Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Yenileme ağacı metodu

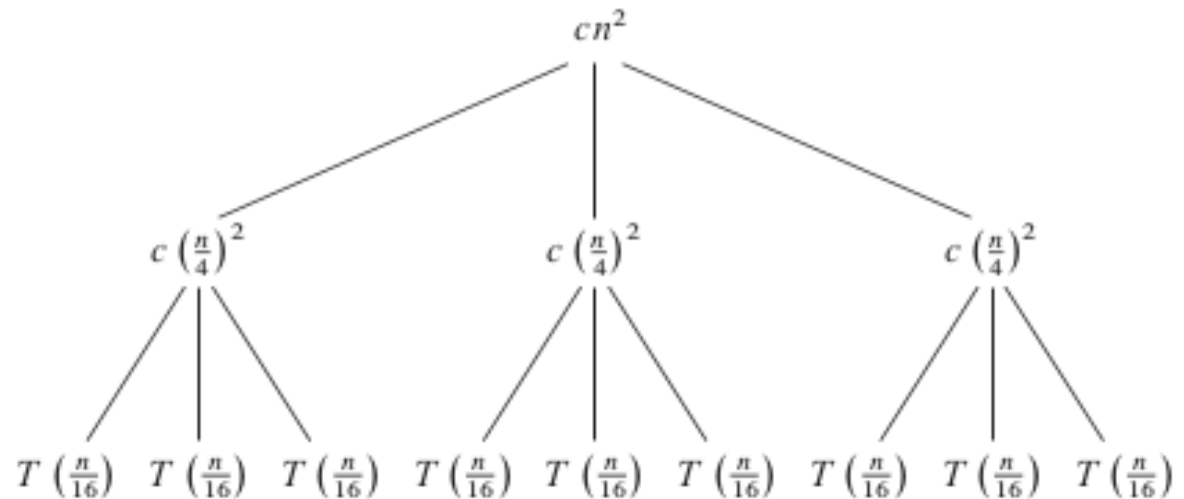
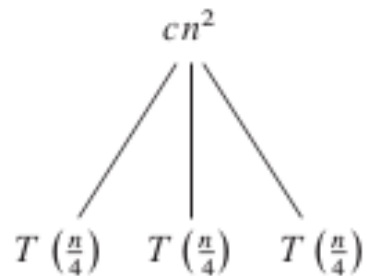
- Değiştirme metodu kullanırken ilk başta doğru olması olası bir tahmin bulmamız gerekiyor.
- Bu tahmini yapmak bazı durumlarda zor olabilir.
- Bu durum söz konusu olduğunda yenileme ağacı oluşturarak tahmini yapabiliriz.
 - Birleştirmeli Sıralama algoritmasını analiz ederken bu yöntemi kullanmıştık.
- Öncelikle her seviyede ne kadar zaman harcanacağı hesaplanır.
- Daha sonra ise kaç tane seviye olacağı bulunur.
- Her seviyede harcanan zamanlar toplam formülleri kullanılarak toplanır ve sonuç elde edilir.

Yenileme ağacı metodu

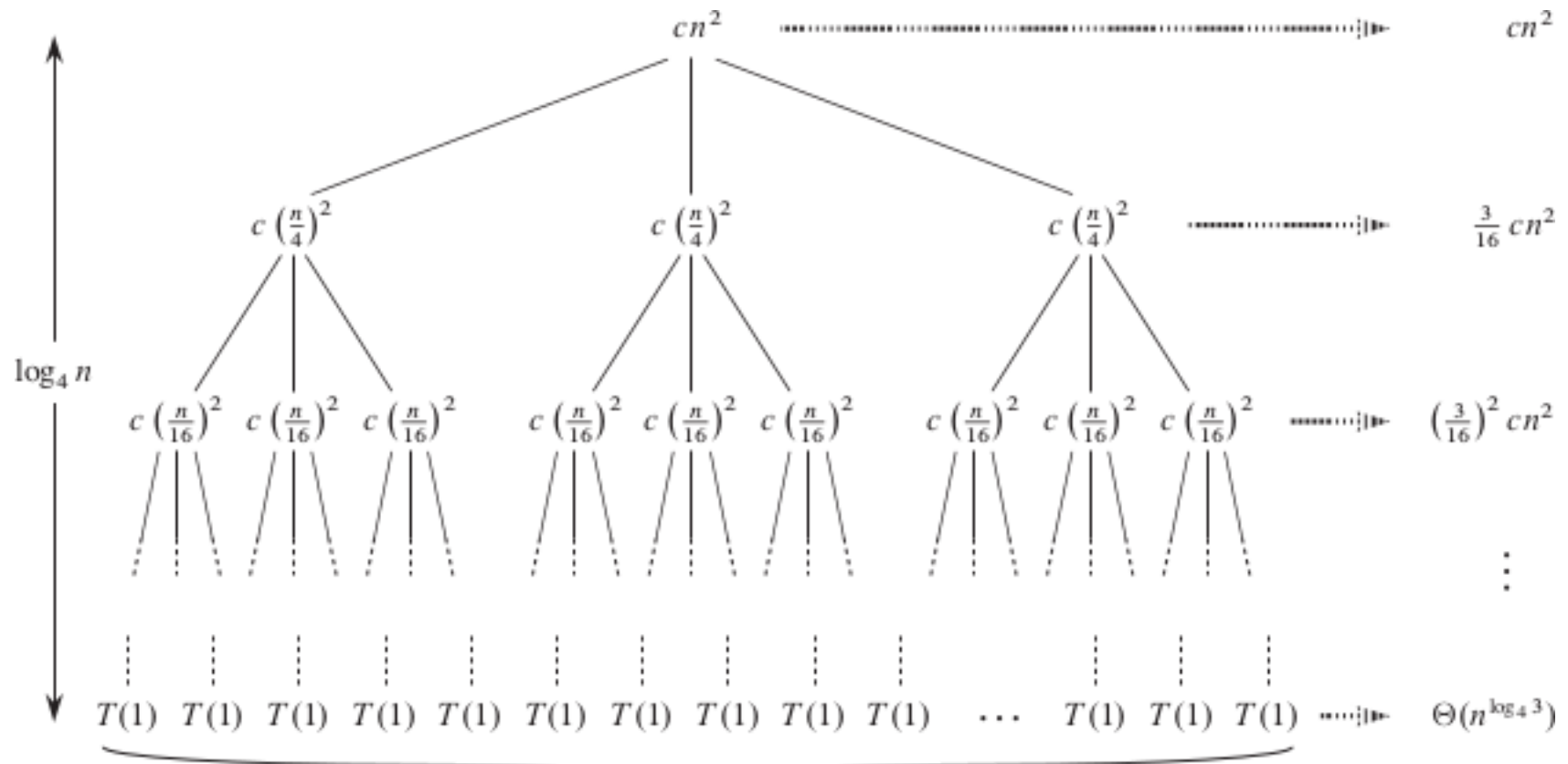
- Örnek: $T(n) = 3T(\lfloor n/4 \rfloor) + \theta(n^2)$
 - Öncelikle taban fonksiyonunu görmezden gelerek $T(n) = 3T(n/4) + cn^2$ için yenileme ağacı çizeceğiz.
 - Geri kalan kısım tahtada anlatılacak.

Yenileme ağacı metodu

$T(n)$



Yenileme ağacı metodu



Master metodu

- Master metot, $T(n) = aT(n/b) + f(n)$, $a \geq 1$ ve $b > 1$ ve $f(n)$ asimtotik olarak pozitif şeklinde yazılan yinelemeli fonksiyonlar için çözüm oluşturmak için kullanılır.
- Bu metodun kullanılabilmesi için üç farklı durum incelenir ve uygun durum bulunursa çözüm kolaylıkla bulunur.
- Yukarıdaki formüle göre problem a farklı parçaya ayrılıyor ve bu parçaların her biri n/b büyüklükte.
- Parçalama işlemi ve bu parçaların birleştirilmesi işlemi $f(n)$ kadar zaman alıyor.
- Master metodu, parçaların büyüklüğe bağlı çözülme süresi ile parçaları oluşturma ve birleştirme sürelerini karşılaştırır.

Master metodu

- Yenileme ağacı düşünüldüğünde üç farklı durum söz konusudur:
 - Çalışma süresi ağırlıklı olarak yapraklar tarafından belirlenir.
 - Çalışma süresi ağacın tamamına düzgün şekilde dağılmıştır.
 - Çalışma süresi ağırlıklı olarak kök tarafından belirlenir.
- Her durum için $f(n)$ ile $O(n^{\log_b a})$ karşılaştırılır.
 - $f(n) = O(n^{\log_b a - \epsilon})$, herhangi bir $\epsilon > 0$ sabiti için. Bu durumda $T(n) = \Theta(n^{\log_b a})$ olur.
 - $f(n) = \Theta(n^{\log_b a})$. Bu durumda $T(n) = \Theta(n^{\log_b a} \lg n)$ olur.
 - $f(n) = \Omega(n^{\log_b a + \epsilon})$, herhangi bir $\epsilon > 0$ sabiti için ve eğer $\alpha f(n/b) \leq c f(n)$ ise, herhangi bir $c < 1$ sabiti ve yeteri kadar büyük her n için. Bu durumda $T(n) = \Theta(f(n))$ olur.

Master metodu

- Örnek: $T(n) = 9T(n/3) + n$
 - Bu durumda $a = 9$, $b = 3$, $f(n) = n$. $n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$
 - $f(n) = O(n^{\log_3 9 - \epsilon})$, $\epsilon \leq 1$ olduğunda, önceki slaytta bahsettiğimiz 1. duruma uygundur.
 - Bu sebeple $T(n) = \Theta(n^2)$.
- Diğer örnekler tahtada gösterilecek.