

MIT Açık Ders malzemeleri

<http://ocw.mit.edu>

6.046J Algoritmalar Giriş, Güz 2005

Bu materyallerden alıntı yapmak veya kullanım şartları hakkında bilgi almak için:

Erik Demaine ve Charles Leiserson, *6.046J Introduction to Algorithms, Fall 2005*. (Massachusetts Institute of Technology: MIT OpenCourseWare). <http://ocw.mit.edu> (Giriş Tarihi: 08, 01, 2010).

Lisans: Creative Commons Attribution-Noncommercial-ShareAlike.

Not: Lütfen atıf yaparken bilgilere eriştiğiniz gerçek tarihi kullanınız.

Atıflar ve Kullanım Koşulları konusunda daha fazla bilgi için:

<http://ocw.mit.edu/terms> ve <http://www.acikders.org.tr> sitesini ziyaret ediniz.

## Ders9

Bugün ikili arama ağaçlarından bahsedeceğiz. Rastgele yapılanmış **ikili arama ağaçları** denen bir şeyden bahsedeceğiz. Ve ikili arama ağaçlarını ders boyunca **BST** olarak kısaltacağım. Sanıyorum hepiniz ikili arama ağaçlarını daha önce özellikle de etütlerde görmüşsünüzdür. Bu nedenle burada bunların ardındaki temel fikirleri geliştireceğiz ve bunları nasıl rastgele hale getireceğimizi ve nasıl iyileştireceğimizi inceleyeceğiz. Aslında bildiğiniz gibi iyi arama ağaçları vardır. bunlar göreceli bir şekilde dengelidir. Şunun gibi bir şeye benzer. Ve bunların yüksekliği  $\log n$  dir. Dengeli dediğimiz zaman, bunlar iyidir.  $\log n$  düzeyinde olan herhangi bir şey iyidir; işe yarar. Arama işlemi bağlamında  $\log n$  düzeyinde olurlar. Bunların dışında kötü arama ağaçları da vardır. Bunların yüksekliği fazladır. Hatta bu yükseklik  $n$  değerine kadar gider. Bu nedenle bu iyidir. Şu da kötüdür. Bizim yapmamız gereken bu ikili ağaçları tasarlarken her zaman iyi yada çoğunlukla iyi olmalarını sağlamaktır. Bu iyileştirme işini yapmanın birçok yolu vardır. Ve eğer problem setini de dahil ederseniz sanırım önümüzdeki birkaç haftada bunların 4 tanesini göreceğiz. Bugün rastgele yapma işlemi kullanarak bunları çoğunlukla dengeli olacak şekilde nasıl yapılandıracağımıza bakacağız. Sonra da problem setinizde bunları daha geniş anlamda iyileştirmeye çalışacaksınız. Bu anlatımı daha ilginç hale getirmek ve sizi motive etmek için bir süre rastgele yapılanmış ikili arama ağacını şimdilik tanımlamayacağım. Ve bu konuyu ilginç hale getirmenin, sizi konuya çekmenin bir yolu da eski arkadaşımız sıralama işlemi kullanmaktır. İkili arama ağaçlarını kullanarak  $n$  kadar sayıyı sıralamanın doğal bir yolu vardır. Size bir A dizilimi verirsem ikili arama ağaçları işlemi bir kara kutu içinde kullanarak bu dizilimi nasıl sıralarsınız? İkili arama ağacını kurarsınız ve ondan sonra da sırayla bunun içinde adım adım gidirsiniz. Kesinlikle doğru. Bu durumda diyelim ki elimizde bir ağacın ilk hali var ve bu boş. Ondan sonra da, bu dizilimin her elemanını bu ağacın içine yerleştireceğiz. Sizin arama ağacını yapılandırmakla ilgili söylemek istediğiniz şey buydu. Böylece  $A[i]$  'yı burada ağacın içine koyalım. Bu bir ikili arama ağacı araya yerleştirmesidir, yani standart bir araya yerleştirmedir. Ondan sonra da bir in-order traversal yani iç **sıralı gezinme** yaparız ki buna kitapta sıralı ağaç yürüyüşü deniliyor. Tamam, siz bu algoritmaların ne olduğunu biliyorsunuz. Ama çabuk bir hatırlatma olması açısından, ağaç araya yerleştirmelerinde temelde  $A[i]$  elemanı aranır. Bunun olması gerektiği yer bulunana kadar aranır. Eğer bunu bulamazsa da o zaman bu değeri yeni bir yaprak olarak olması gereken yere yerleştirir. Ağaç yürütmesinde öz yinelemeli olarak sol alt ağaçta yürünür. Bunu köke kadar sürdürür ve ondan sonrada sağ alt ağaçta öz yinelemeli yürüyüş başlar. Ve ikili ağaç aramalarının bir özelliği nedeniyle bu elemanları sıralı bir şekilde çıkışa verir. Haydi bu konuda bir örneği hızlıca yapalım. Aslında bu örnek sizin daha önce gördüğünüz bir sıralama algoritmasıyla da ilgili olacak. Örnek her ne kadar küçük bir örnekse de burada göreceğimiz ilişki aslında oldukça şaşırtıcıdır. Aslında benim için bu böyleydi, bu dersi vermeye ilk başladığım zamanlarda..

Evet dizilimim 3, 1, 8, 2, 6, 7, 5. Ve bu elemanlara soldan sağa doğru sırayla erişeceğim ve ağacımı kuracağım. Bu durumda gördüğüm ilk eleman 3. Ve bu 3'ü boş ağaca yerleştiriyorum. Bunun herhangi bir karşılaştırmaya ihtiyacı yok. Sonra 1'i araya yerleştireceğim. Bakıp karşılaştırıyorum: 1, 3'den küçük müdür büyük müdür? Küçüktür. Bu nedenle bunu buraya koyuyorum. Sonra 8'i araya yerleştiriyorum. Bu 3'den büyüktür. Bu nedenle buraya yeni bir yaprak eklenir.

Sonra 2'yi araya yerleştiririm. Bu 1 ile 3 arasında bir yerdedir. Bu nedenle bunu 1'in sağ ardılı olarak yerleştiririm. Bu nedenle 2'yi şuraya koyarım. 6, 3'den büyüktür, ama 8'den de küçüktür. Bu nedenle 6 buraya gider. 7, 3'den büyüktür ve 8'den küçüktür; 6'dan da büyüktür. O nedenle buraya gider. Ve 5, 3 ile 6 arasındadır. Onun için şuradadır. Ve bu şekilde bizim ikili ağacımız ortaya çıkmış olur. Ondan sonra da bir iç sıralı gezinme yaparım ve bu şekilde bunun sonucunda 1, 2, 3, 5, 6, 7 ve 8'i çıktı olarak alırım. Şimdi büyükçe bir yığının olduğu için bunu kafamda şöyle bir gözden geçirebilirim. Ama bu noktada biraz dikkatli olmam gerekiyor. Doğal olarak yapmanız gereken şey bunların sıralı bir düzende çıkıp çıkmadığıdır: 1 2 3 5 6 7 8. Ve eğer büyük bir kümeniz yoksa o zaman gidip bir tane alabilirsiniz. Bu her zaman yararlıdır. Bellek fiyatları bugünlerde yükseliyor mu yoksa aşağıya mı gidiyor? Aslında bunu tanımlayan şey politika. Fiyat kontrolü yada her neyse.

Şimdi soru, algoritmanın koşma süresi nedir? Bunun yanıtı aslında değişken olduğudur. Çözümlemesi kolay olan kısım için başlangıç yapılanmasıdır. Düzenli ağaç yürümesi, bu ne kadar zaman alır? n, güzel. Bu durumda yürüyüş için n düzeyinde ve başlangıç için de bir sabit olacaktır. Buradaki soru n sayıdaki ağaç araya yerleştirmelerini yapma sürem ne kadar olacaktır? Bu sorunun yanıtını "belli olmaz"dan başka şekilde cevaplandırmak isteyen kimse var mı? Oradaki büyük kargaşayı aslında ben çözdüm. Evet. Büyük Omega ( $n \log n$ ). Evet bu doğru. Bu en azından  $n \log n$  dir. Neden? ... Doğru. Yani siz bana iki yanıt verdiniz. Birincisi bu karar ağacının alt sınırlamasına bağlıdır. Ama bu durumu aslında kanıtlamaz. Burada biraz daha dikkatli olmanız lazım. Burada savunulan her zaman  $\Omega(n \log n)$  olacağıdır; en kötü durumda  $\Omega(n \log n)$  olduğu kesindir. Her karşılaştırma tabanlı sıralama algoritması en kötü durumda  $\Omega(n \log n)$  dir. Ama bu her seferinde  $n \log n$  olacaktır. Aslında  $\Omega(n \log n)$  dir, çünkü bu ikinci yanıtınıza bağlıdır. Yani mükemmel dengeli bir ağaç olması alınabilececek en iyi sonuçtur. Şimdi tahtaya çizdiğim bu şekil hayatımda en çok çizdiğim şekildir. Sanırım 15 düğümlü bir mükemmel ağaç. Yani şanslıysak bunu elde ederiz. Ve buradaki boğumların derinliklerini toplayacak olursanız bu size ağacı aramanın maliyetini verir. Ve özellikle de şu en alttaki  $n/2$  boğumlarının derinliği  $\log n$  dir. Bu nedenle hepsi için en azından  $n \log n$ 'yi harcamanız gerekir. Eğer dengeli değilseniz o zaman bu daha da kötü olacaktır. Bunun kanıtlanmaya ihtiyaç vardır ama bu doğrudur. Yani her zaman  $\Omega(n \log n)$  değerini alır.

Ama başka durumlar da olabilir; elemanlarınızın sıralı olduğunu biliyorsanız bunu doğrusal sayı karşılaştırmalarıyla çözebilirsiniz, ama burada bunu yapamazsınız. Bu soruyla ilgili başka tahmini olan var mı? Evet? Büyük  $O(n^2)$ . Güzel. Neden? Doğru.

Çünkü biz  $n$  işlem yapıyoruz ve her düğümün yüksekliği en azından  $n$ . Yani araya yerleştirdiğimiz eleman başına yaptığımız karşılaştırma sayısı en çok  $n$ . Bu nedenle de bu en fazla  $n^2$  olur. Başka yanıt var mı? Bu algoritmanın  $n^2$  zamanını alma olasılığı var mı? Bu değer  $n^2$  olması için bazı durumlar olabilir mi? Bu eğer sıralı bir haldeyse bu kötü olacaktır. Yani bu zaten sıralanmışsa veya tersten sıralıysa o zaman kötü bir durumda olursunuz. Çünkü şuna benzer bir ağaç elde edersiniz. Bu sıralanmış haldir. Ve sonra hesaplırsınız. Böylece ağaçtaki toplam zaman için, tüm düğümlerin derinliğinin toplamını  $x$  değiştikçe alırsınız. Durumumuzda bu  $1+2+3+4$  yani aritmetik bir seri olur. Bunlardan  $n$  tane var o zaman da Teta  $n^2$  olacaktır; yani  $n$  kare bölü 2 gibi bir şey. Ve bu kötü haberdir. Bu algoritmanın en kötü durumda koşma süresi  $n^2$  olur. Bu size bir yerden tanıdık geliyor mu? Yani algoritmaların en kötü durumda koşma süresinin  $n^2$  olduğu sıralanmış bir durumda? Ama eğer şanslıysak yani şanslı durumda o zaman zaten bu sıralanmış dengeli bir ağaçtır. Bu çok iyi olmaz mı? Omega  $\log n$  yüksekliğindeki herhangi bir şey bize  $n \log n$  zamanında çalışan bir sıralama algoritması verir.

Yani şanslı durumda  $n \log n$  yi elde ediyoruz. Ama şanssız durumda  $n^2$  elde ediyoruz ve bu şanssızlık sıralanmış olmaktan kaynaklanıyor. Bu size daha önce gördüğünüz bir algoritmayı hatırlatıyor mu? Çabuk sıralama. Bu algoritmanın koşma süresi çabuk sıralamanın koşma süresine sıkı sıkıya bağlıdır. Öyle ki bu algoritmanın yaptığı karşılaştırmalar çabuk sıralamanın yaptığı karşılaştırmalarıyla tıpatıp aynıdır. Bunu başka bir sırada yapar ama aslında aynı algoritmanın bir şekil değiştirmiş hali gibidir. Buradaki sürpriz de bu.

Daha önce çabuk sıralamayı analiz etmiştik. Burada da o çözümlemenin yan faydasını bedelsiz olarak elde edeceğiz. Buradaki ilişki BST sıralamasıyla çabuk sıralamanın aynı karşılaştırmaları başka bir düzende yapmalarıdır. Şimdi daha önce kullandığımız örneği bir kez daha ele alacağım: 3, 1, 8, 2, 6, 7, 5. Şimdi burada bir dizilim var. Bu dizilimin üzerinde çabuk sıralamanın belli bir türünü çalıştıracamız. Burada biraz dikkatli olmam gerekiyor. Aslında çabuk sıralamanın hemen göze çarpan bir versiyonu olacak. Hatırlarsanız uygulamada standart olarak o sıkıcı çabuk sıralamada ilk elemanı bölüntü elemanı olarak alıyordunuz. Bu nedenle burada 3'ü alacağım. Ve sonra 3'ten küçük olan elemanları yani 1 ve 2'yi bölüntünün bir bölümüne alacağım. Ve 3'ten büyük olan elemanları yani 8, 7, 6 ve 5'i alacağım ve çabuk sıralamanın bu türünde 8, 6, 7 ve 5 elemanların sıralarını değiştirmeyeceğim. Burada sıranın değişmeyeceğini sıranın korunacağını söylememiz lazım aksi takdirde eşitlik tutmaz. Sonuçta bu bir tür dengeli bölüntü algoritmasıdır. Ama oldukça kolaydır. Bu çabuk sıralamanın belirli bir türüdür. Ve yakında bunu rastgele hale getireceğiz. Ve bunu rastgele hale getirdikten sonra da farklılık ortadan kalkacak. Tamam şimdi bölüntü elemanımızda yarattığımız soldaki özyinelemeye bakalım. Burada 1 den küçük olan şey, 0 yada yok. 1 den büyük olan şeyler var ki bu değer 2. Sonra da bizim bölüntü elemanımız geliyor. Burada işimiz bitti. Bu tarafta da 8 ile bölüntüyü yapacağız. Burada her şey 8 den küçük; bu durumda 6, 7, 5 elde ediyoruz ve sağ tarafta da bir şey kalmıyor. Sonra 6 ile bölüntüyü uygulayacağız. 6 dan küçük

olan deęer 5 ve 6 dan byk olan deęer de 7. ve bunlar da aslında blnt elemanları ama abuk yoldan elde edildiler. Őimdi blnt elemanlarıyla elde edilen bu aęa Őu aęaca mthiŐ benziyor. Aslında aynı aęa olması gerekiyor. Adım adım gidersek, abuk sıralama hangi karŐılaŐtırmaları yapıyor? Aslında nce her Őeyi 3'le karŐılaŐtırıyor; kendisi dıŐında her Őeyi. Őimdi Őuraya bakarsanız yeni elemanları araya yerleŐtirdięimizde ne oluyordu? Olan Őey her seferde bir eleman yerleŐtirdięimizde ilk yapılan bunun 3 ile karŐılaŐtırılması. Eęer 3 den kkse o zaman sol dallanmaya gidiyoruz; eęer 3 den bykse o zaman saę dallanmaya gidiyoruz. Yani 3 ile olan tm bu karŐılaŐtırmaları her iki durumda da yapıyoruz. Sonra eęer 3 den kk bir eleman varsa, bu ya 1 yada 2 dir. Eęer 1'se iŐimiz bitiyor. Burada bir birle karŐılaŐtırılmıyor ama 2'yi 1 ile karŐılaŐtırıyoruz. Aslında 2'yi araya yerleŐtirirken 3'le karŐılaŐtırıyoruz ama aynı zamanda 1'le karŐılaŐtırıyoruz. Sonra burada da benzeri oluyor; aynı abuk sıralamada olduęu gibi. 3 den byk elemanların durumunda biz her Őeyi 8 ile karŐılaŐtırıyoruz. nk burada 8'e gre bir blnt yaptık, Őurada ise 3'ten sonra gelen boęum 8. 8'i araya yerleŐtirdięimizde de her Őeyin 8 den kk olup olmadıęını karŐılaŐtırıyoruz ve byle gidiyoruz; yani yaptıęımız karŐılaŐtırmaların hepsi aynı ama sıraları biraz farklı. Yani 90 derece dnyoruz. Aslında hoŐ bir durum. Ve bunun zmlemede bazı sonuları ortaya ıkıyor. zellikle en kt durum halinde yrtm sresi Teta  $n^2$  oluyor ve bu da ok heyecan verici deęil.

Bizi burada gerekten ilgilendiren rastgele dzenlenmiŐ hali . nk bu durum iyi sonu verir. Rastgele hale getirilmiŐ BST sıralaması rastgele abuk sıralamanın ok benzeridir. Sonuta yaptıęınız ilk Őey dizilimi tek biimli ve rastgele olarak deęiŐtirmek ve btn permtasyonları eŐit olasılıklı hale getirmektir. Sonra da BST sıralamasını alıŐtırırız. Bu temelde rastgele abuk sıralamanın formle edilmiŐ bir halidir. Bundan sonrada rastgele BST sıralaması rastgele abuk sıralamayla aynı karŐılaŐtırma iŐlemlerini yapar. Burada kk rastgele seiyoruz ve Őuradaki abuk sıralamada da blnt elemanını rastgele seiyoruz. İkisinde de aynı farklılık var. Bu nedenle de bu algoritmanın sresi rastgele abuk sıralamanın aynıdır. nk her iki 2 durumda da aynı karŐılaŐtırmaları yapıyoruz ve yapılan karŐılaŐtırmanın sayısı eŐit; bu aslında rasgele deęiŐkenler iinde byledir. Bu algoritmanın rasgele deęiŐkeni ve yrtm sresi Őu algoritmanın rasgele sresine ve rasgele deęiŐkenine eŐittir; zellikle de umulanlar birbirine eŐittir. Ve bu durumda biz rasgele abuk sıralamanın n elemanla beklenen koŐma sresini biliyor muyuz? Evet.  $O(n \log n)$ . İyi burada biraz endiŐelenmiŐtim. Tamam zellikle BST' nin beklenen koŐma sresi  $n \log n$ . Tabii bu sıralama baęlamında ok heyecan verici deęil. Sıralamayı burada bu iliŐkiyi ortaya ıkarmak iin kullandım. Burada bizi ilgilendiren Őey ve BST sıralamasını anlatmamın nedeni aęacın nasıl grneceęidir. Bizim aslında istedięimiz bir arama aęacıdır. Arama aęacı sıralamadan daha fazla Őeyler de yapar. n dzeyindeki adımlamalar bir arama aęacında yapılacak sıkıcı iŐlemlerden biridir. Aramayı bir arama aęacında yaparsınız; bu da ok heyecan verici deęildir. Aslında elemanları nce sıralarsınız, sonra da bunları bir dizilime yerleŐtirip ikili arama yaparsınız. Ama ikili arama aęalarıyla ikili arama dizilimlerini karŐılaŐtırdıęınız zaman, aęaların farkı

dinamik olarak güncellenebilmeleridir. Bu işlemi bugün bu derste yapmayacağız daha sonraki problem setlerimizden bir tanesinde ele alacağız. Şu anda sadece ısınma turlarını atlıyoruz. Diyelim ki elemanlar değişmiyor. Bir ağacı en baştan yapılandırıyoruz.  $n$  sayıda tüm elemanlarımız işin başında elimizde var. Ve biz bunu rastgele olarak yapılandıracağız. Yani bu dizilimi rastgele olarak devşireceğiz, sonrada tüm elemanları bir ikili arama ağacının içine atacağız. BST sıralamasının yaptığı iş budur. Sonra  $n$  düzeyinde bir adımlama devreye girer ama bu  $n$  düzeyindeki adımlama aslında beni ilgilendirmez. Beni ilgilendirmiyor çünkü zaten bunu çözümlemiştik. Ama eğer işlem bitmiş olsaydı bu ders çok kısa olurdu. İstedığımız şey aslında bir BST'yi rasgele yapılandırmak ve bu algoritmanın yaptığı işte budur. Bu ağaç rastgele BST sıralamasından ortaya çıkar, yani basit ağaç araya yerleştirme algoritmasını kullanarak bu elemanların oluşturduğu dizilimi rastgele devşiririz. Buradaki soru bu ağaç neye benzeyecek? Ve özellikle bundan gerçekten çıkarabileceğimiz bir sonuç var mıdır? BST sıralamasının beklenen koşma süresi  $n \log n$ . Dikkat ettiyseniz şu ana kadar BST sıralamasının koşma süresinin ne olduğunu birkaç kez tekrarladım. BST sıralamasının  $n$  eleman üzerinde toplam süresi, tüm düğümler yani  $x$  değerleri değiştiğinde düğümlerin derinliklerinin toplamına eşittir. Derinlik sıfırdan başlar ve kök elemanına kadar gider, onun ötesinde başka bir karşılaştırma yapmazsınız çünkü kök elemanında karşılaştırma yapmazsınız; tüm karşılaştırmalar derinlik her neyse o boyutta olur. Bu durumda da buradakinin beklenen cinsinden  $n \log n$  olduğunu biliyoruz. Bu bize ağaç hakkında ne söyler? Yani ağaçtaki tüm düğümler, yani  $x$ 'ler konusunda? Örneğin bu bize ağacın yüksekliği hakkında bir bilgi verir mi? Evet? Doğru sezgisel olarak bu bize ağacın yüksekliğinin  $\Theta(\log n)$  olduğunu ve  $n$  olmadığını söyler. Ama gerçekte bunu bize göstermez. Ve bunu sadece sezgisel olarak ele aldığınızda bunun doğru olmayabileceğini de düşünmeniz lazımdır. Aslında doğru da değildir. Size bunun gerçekte ne olduğunu anlatayım. Şimdi eğer her iki tarafta da bekleneni ele alırsak burada  $(n \log n)$  elde ederiz. Yani buradaki beklenen değer  $(n \log n)$  dir. Şimdi burada ise beklenen toplam derinliği elde ederiz ki bu çok heyecan verici bir şey değildir. Şimdi beklenen ortalama derinliğe bir bakalım. Şimdi  $(1/n)$ 'ye bakacak olursam bu ağaçtaki tüm  $n$  düğümün  $x$  derinliğindeki toplamları alındığında bu tüm düğümler üzerindeki ortalama derinliktir.

Burada elde etmem gereken  $\Theta(n \log n) / n$  'dir çünkü her iki tarafı da  $n$  ye böldüm. Burada kullandığım beklenenin doğrusallığıdır ve bu da  $\Theta(\log n)$  ye eşittir. Böylece burada karşılaştığım gerçek, beklenen koşma süresinin bana ağacın ortalama derinliğinin  $\log n$  olduğunu söylediğidir; ama bu aslında ağacın yüksekliğinin  $\log n$  olduğu anlamına gelmeyebilir. Şimdi hatırladığınız gibi ağacın yüksekliği herhangi bir düğümün maksimum derinliğiydi. Burada biz ortalama yüksekliği sınırlıyoruz.

Şimdi ağaçla ilgili bir örneğe bakalım; yine en çok sevdiğim resmi çizeceğim. Şimdi burada hoş bir dengeli ağaç var diyelim ki sevdiğim ağacın düğümlerinin yarısı kadar veya birazcık fazla. Sonra da bir yapraktan uzayan oldukça uzun bir yol var. Hangisinden olduğu önemli değil. Ve diyeceğim ki bu yolun toplam uzunluğu karekök  $n$ 'dir ve bu da  $\log n$ 'den oldukça büyüktür. Bu yaklaşık  $\log n$ 'dir. Aslında bu,

$\log(n - \sqrt{n})$  gibi bir şey olur. Yani bu düğümlerin logaritmik yükseklikleri, pardon logaritmik derinlikleri vardır. Bu ağaçta ortalama derinliği hesaplayacak olursanız bir çok düğümün, örneğin  $n$  düğümün yüksekliği ( $\log n$ ) olsun; aşağıda geri kalan en çok kök  $n$  sayıda düğümün derinliği de en fazla kök  $n$  olsun. Yani bu en çok kök  $n$  çarpı kök  $n$  olur. Aslında bakarsanız bu yaklaşık şunun yarısı kadardır ama bu çok önemli değildir. Yani bunlar  $n$  eder. Böylece bu  $n \log n$  olur, af edersiniz ortalama derinlik için: her şeyi  $n$ 'ye bölmek zorundayım.  $n \log n$  aslında ortalama yükseklik ya da ortalama derinlik olarak biraz büyük kalacak. Bu durumda buradaki ortalama derinlik ( $\log n$ ) dir, ama ağacın yüksekliği ( $\sqrt{n}$ ) dir. Yani bu yeterli değildir. Ortalama derinliğin  $\log n$  olduğunu bilmek yüksekliğin  $\log n$  olduğu anlamına gelmez. Ama bugünkü teoremin anlatmak istediği şey rastgele yapılanmış bir ikili arama ağacının ortalama yüksekliğinin gerçekten de  $\log n$  olduğudur. BST =  $O(\log n)$  düzeyindedir. Bilmek istediğimiz de zaten bu, çünkü bu bize diyor ki bir ikili arama ağacını rastgele yapılandırırsak o zaman bunun içindeki aramayı  $\log n$  sürede yapabiliriz. Bu, sıralama açısından o kadar önemli değildir. Biz bu ağacı oluşturmanın beklenen koşma süresiyle ilgileniyoruz. Ve diyoruz ki bu teoremi kanıtlayabilirsek aramayı çoğunlukla hızlı bir beklenen sürede yapabiliriz. Onun için bugünkü dersin geri kalan süresini bu teoremin kanıtlanmasına ayıracağız. Hayal edebileceğimiz gibi bu oldukça şaşırtıcı olacak. Bu çabuk sıralama ya da benzeri durumlarda önemli bir olasılık çözümlemesi oluşturacak.

Şimdi eğer teoremlerle ilgili başka sorular yoksa ben kanıtlamanın çerçevesiyle işe başlayacağım. Ne kanıtlamak istediğimiz aslında oldukça açık olmalı. Şu ana kadar gördüğümüz çözümlemelerin hepsinden daha garip bir çözümleme olacak. Ve burada bir rastgele değişkenin üstelini almayı kullanarak güzel bir numara yapacağız. Ve bunu yapabilmemiz için de **Jensen'in eşitsizliği** denilen bir gereç kullanmamız gerekiyor. Şimdi bu gereci yada aracı kanıtlayacağız. Genelde olasılık araçlarını ya da yöntemlerini kanıtlamayız. Ama bunu kanıtlayacağız. Çok zor bir iş değil. Ayrıca temel bir çözümlemedir. Şimdi önkuram diyor ki; eğer elimizde dışbükey bir fonksiyon  $f$  varsa ve bunun ne demek olduğunu hepimizin bilmesi gerekiyor ama eğer unuttuysanız bunun tanımını birazdan yaparım; Eğer elinizde dışbükey yada konveks bir fonksiyon  $f$  ve rastgele bir  $x$  değişkeni varsa, beklenen  $f$ 'si, en çok o rastgele değişkenin  $f$ 'sinin beklenendir. Bununla ilgili yeterince düşünün ve sezgisel olarak dışbükeye yakın bir fonksiyonu çizin. Ama biz bunu kanıtlayacağız. Bunun bize sağlayacağı kolaylık ağacın yüksekliğini gösteren rastgele değişkeni çözmek yerine, burada  $X_n$ 'ye rastgele değişken diyelim. (r.v.) yani rastgele değişkenin kısaltması; BST'nin  $n$  sayıda düğümde yüksekliğinin r.v.'sı =  $X_n$ . Bu çözümleneceğimiz yapılanma olacak.

Şimdi bu istenen rastgele değişken olan  $X_n$ 'yi çözmek yerine, pardon buradaki  $X$  büyük harfle yazılmalıydı.;  $X_n$ 'nin herhangi bir dışbükey fonksiyonunu çözümleriz. Ve burada bunun üstel halini çözümleneceğiz. Şimdi  $X_n$ 'yi, ikinin  $X_n$ 'ninci kuvveti olarak tanımlayacağım. Şimdi buradaki önemli soru bunu yapmakla

niye uğraştığımızdır. Yanıt; çünkü bu çalışır ve eğer  $X_n$ 'yi çözümlenmeye kalksaydık o zaman çalışmayacaktı. Bununla ilgili sezgisel bir şeyleri ilerde göreceğiz ama çok da sezgisel olmayacak. Buradaki çözümlenmemizde bu ek kurnazlığa ihtiyacımız var. Önce  $Y_n$  'nin beklenenini sınırlayacağız ve o noktadan sonrada Jensen'in eşitsizliğini kullanarak  $X_n$  'nin beklenenini konusunda bir sınır elde edeceğiz; bu oldukça katı bir sınır olacak. Bunun nedeni, eğer üsteli belli sabit faktörler boyutunda sınırlayabilirsek bu durumda  $X_n$  'yi daha iyi sınırlayabiliriz çünkü  $X_n$  'yi elde etmek için logaritma kullanırız. Burada aslında sabitin de ne olacağını hesaplayacağız. Şimdi bu kanıtlamanın özünde olan şey  $Y_n$  'nin beklenen süresinin  $n^3$  düzeyinde olduğudur. Burada sabitin ne olduğunu gerçekte bilmiyoruz. Ve buna ihtiyacımız da yok. Sonra bu parçaları birleştireceğiz. Haydi şimdi bunu yapalım. Bizi ilgilendiren şey  $X_n$  nin beklenenini yani ağacımızın yüksekliğidir. Bulacağımız şeyler bununla ilgili olacaktır. Şimdi şurada yatay boyutta biraz boşluk bırakın. İkinci  $X_n$  kuvveti düzeyinde bir beklenen elde ederiz. Bu  $Y_n$  'nin beklenenidir. Ve bunun da  $n^3$  düzeyinde olduğunu öğrendik. Jensenin eşitsizliği bize der ki; eğer bu  $2^X$  fonksiyonunu alırsak ve bunu şuraya uygularsak, sol tarafta ikinci  $E[X_n]$  'ninci kuvvetini elde ederiz. Yani biz ikinci  $E[X_n]$  'ninci kuvvetinin, en çok ikinci  $X_n$  'ninci kuvvetinin  $E$  ' si olabileceğini elde ederiz. Şimdi elimizde bir sınır var. Bu durumda deriz ki ikinci  $E[X_n]$  kuvveti en çok  $n^3$  dür ve her iki tarafta da logaritmasını aldığımızda  $E[X_n]$  in en çok  $n^3$  ün logaritması olduğu sonucu çıkar. Şimdi ben bunu komik bir şekilde yazacağım ve diyeceğim ki  $n^3$  düzeyinin logaritması bize sabiti verecek. Bu  $3 \log n + O(1)$  olacaktır. Böylece rastgele yapılanmış bir ikili arama ağacının beklenen yüksekliğinin  $n$  düğüm üzerinde kabaca en çok  $3 \log n$  olabileceğini kanıtlayacağız. Bununla ilgili ileride daha fazla şey söyleyeceğim. Şimdi siz çözümlenmenin sonunu gördünüz. Tabii bu bir ön ima ya da ön belirtidir. Şimdi bu yukarıdan aşağıya yaklaşımdır. Buradaki adımların ne olduğunu şimdi görüyorsunuz. Şimdi buradaki adımları teker teker ele almamız gerekir. Adım1: Bir parça uğraştırır ama kolaydır, çünkü basit temel şeylerdir. Adım2: Sadece bir tanımlama yaptık ve işimiz bitti. Adım3: Herhalde en zor kısmıdır. Adım 4: Zaten yapmıştık. Şimdi birinci adımla başlayalım.

İlk yapmamız gereken şey dışbükey bir fonksiyonu tanımlamaktır, çünkü tanımlama ile oldukça fazla uğraşacağız. Şimdi bu gerçek bir çözümlenmeden çıkan bir kavram olacak; eğer ilgili çözümlenme dersini almadıysanız, burada çözümlenme calculus yerine kullanılacak takma sözcük olacak. Calculus dersinde dışbükeyliğin ne olduğunu görmüş olmanız gerekirdi. Bir dışbükey fonksiyon şuna benzer bir şeydir. Tamam, iyi oldu. Bu kavramı resmi hale getirmenin bir yolu eğri üzerindeki iki noktayı ele almaktır. Bu evrede sadece gerçek değerleri ele alan fonksiyonlarla ilgileniyorum Onun için şöyle görünecektir: Bu bir şeylerin  $f$  'sidir; ve burası da o bir şeylerdir. Bu eğrinin üzerindeki iki noktayı aldığımızda ve bu iki nokta arasında bir doğru çizdiğimizde, bu doğru her zaman o eğrinin üzerinde bir yerde olur. Dışbükeyliğin tanımı budur. Bunun geometrik bir kavramı da vardır ama temelde aynıdır. Ama fonksiyonlar



açısından bu doğru diliminin eğrinin üzerinde yer alması gerekir. Bu parçanın dışındaki doğru parçaları, eğrinin üzerinde yer almaz. Eğer bunu daha da uzatırsam o zaman eğrinin altına doğru geçecektir doğal olarak. Ama bu dilimde parça eğrinin üzerinde kalmalıdır.

Şimdi bunu biraz daha tanımlanır bir hale getireceğim. Buna  $x$  diyeceğim. Böylece şu  $f(x)$  olacak. Ve buna da  $y$  diyeceğim bu da  $f(y)$  olacak. Buradaki iddiam,  $x$  ile  $y$  arasında herhangi bir sayıyı alıp baktığımda, eğrinin üzerindedir diyebilmektir. Bu noktanın karşılığı doğru parçamda şuradadır. O noktanın  $y$  değeri bu durumda şuradaki  $y$  değerinden daha büyük yada ona eşit olmalıdır, değil mi? Bu noktanın hesaplanmasında biraz geometriye ihtiyacımız olacak. Aynı zamanda eminim ki bu bir çözümlenme konseptidir. Ama ben geometriye daha yakın olduğum için buna geometri diyorum. Eğer elinizde  $p$  ve  $q$  gibi iki nokta olursa ve bunların arasındaki doğru parçasını parametrize etmek isterseniz, ki ben burada şu 2 nokta arasında parametrize etmek istiyorum. Bunu yapmanın yolu bir doğrusal kombinasyon yada birleştirmedir. Eğer biraz doğrusal cebir biliyorsanız, doğrusal birleşim şöyle bir şeye benzer. Burada affine combination yada ilgin birleşim denen bir kavramı ele alacağız. Ve bu kavram çerçevesinde  $\alpha + \beta = 1$  olur. Bunun sonucunda böyle noktaları ele aldığınızda  $\alpha$  gibi bir sayıyı  $p$  noktasıyla çarparsınız, buna  $\beta$  gibi bir sayıyı  $q$  noktası ile çarparsınız ve elinizde  $\alpha + \beta = 1$  vardır; eğer bütün bu noktaları alırsanız o zaman buradaki bütün çizgiyi elde edersiniz ve bu biraz karışıktır. Ama biz tüm çizgiyi istemiyoruz. Eğer alfa ve beta'nın negatif olmaması sınırlamasını getirirseniz o zaman şu doğru parçası elde edilir. Böylece bu  $\alpha$  ve  $\beta$ 'yi 0 ile 1 arasında değerler almaya zorlar. Çünkü bunların toplamı zaten birdir ve tabii negatif değillerdir. Bu nedenle burada yapacağımız şey  $\alpha x + \beta y$  yi ele almaktır. Bu bizim şu 2 sınır arasındaki noktamız olacaktır:  $\alpha + \beta = 1$  ve  $(\alpha + \beta) \geq 0$ . Böylece bu nokta şunun  $f$  si haline gelir. Bu  $f(\alpha x + \beta y)$  olur. Ve bu nokta da  $f_x$  ile  $f_y$  arasındaki doğrusal interpolasyon. yada içkestirimdir ki, aslında ötekiyle aynıdır. Yani  $\alpha f(x) + \beta f(y)$  dir. Tamam buraya kadar sezgiseldi. Eğer bunu tam anlamıyla takip edemediyse çok da üzülme, çünkü burada ilgilendiğimiz şey bir takım olguları kanıtlamakta sembolik bir yanıt almaktır. Ama bunun geldiği yer budur. Şimdi tanım yapacağım. Bunun fonksiyonu dış bükey.

$\mathbb{R}$  içindeki tüm  $x$  ve  $y$  değerleri için,  $\alpha$  ve  $\beta \geq 0$  olması ve  $\alpha + \beta = 1$  olması durumlarında:  $f(\alpha x + \beta y) \geq \alpha f(x) + \beta f(y)$  'den daha küçüktür yada ona eşittir. Bu sadece buradaki  $y$  koordinatının şu  $y$  koordinatından daha küçük yada ona eşit olması demektir. Ama bu, şeklin arkasındaki sembolizmadır. Şimdi Jensen'in eşitsizliğini kanıtlamak istiyoruz. Aslında daha tam da orada değiliz. Önce basit bir ön kuramı kanıtlayacağız, sonra Jensen'in eşitsizliğini kanıtlamak daha kolay olacak. Şimdi kanıtlayacağımız şey şudur: Burada dış bükey fonksiyonlarıyla ilgili ön kuram var. Bunu daha önce görmüş olabilirsiniz. Ama bu Jensen'in Eşitsizliği için çok önemli olacak. Bunun 2 şey yerine  $n$  şeyin ilgin birleşimleri ile ilgili bir tanımlama olduğunu düşünün. Farz edin ki elimizde  $n$  adet gerçek sayı var. Ve  $n$  tane  $\alpha_i$  değerimiz var. Ve bunlar  $\alpha_1$ 'den  $\alpha_n$  ye kadar devam ediyor. Bunların hiç biri negatif değil. Ve bunların

toplama da 1 e efit. Yani  $\alpha_k$  nin toplamlarının  $k=1$ 'den  $n$ 'ye kadar deęiřtięinde 1 e efit olduęunu s6ylyorum. Evet , varsayımlarımız bunlar. Sonu aynı Őey ama biz t6m  $k$ 'ları hesaba katarak bunu toplayacaęız. B6ylece  $k=1$ 'den  $n$ 'ye kadar olan bir deęiřimde ( $\alpha_n$  arpı  $x_n$ ) nin toplamlarının  $f$  fonksiyonunun karřısında alfalar arpı  $f$  lerin  $k=1$ 'den  $n$ 'ye kadar artarken olan toplamları olacak. Dıřb6keylięin tanımı Őuradaki tanımlamadır ama burada  $n=2$  durumu vardı. Burada  $\alpha_1$  ve  $\alpha_2$ , beta'dır. Bu aslında genel  $n$  iin bir varsayımdır. Ve bunu daha komik bir Őekilde yorumlayabilirsiniz ama ben bu iře girmeyeceęim. Hayır, niye girmeyeyim? Ben bir geometriciyim. Bu, bu eęrinin 6zerinde birkaç noktayı aldınız diyelim.

Bu noktaların tanımladıęı poligonu alırsınız. Bunlar doęru paralarıdır. Bunun iini alırsınız. Bunun gibi bir ilgin birleřim oluřturursanız, poligonun iindeki veya sınırındaki bir noktayı elde edersiniz. Buradaki iddia b6t6n bu noktaların eęrinin yukarısında olduęudur. Tekrar ediyorum sezgisel olarak g6zel kanonik bir dıřb6key eęri izerseniz bu doęrudur ama aslında cebirsel olarak da doęrudur. Bu her zaman iyi bir Őeydir. Bu teoremi yada bu 6nkuramı nasıl kanıtlayacaęımız konusunda 6neriler var mı? Aslında ok kolay. Bunu kanıtlamak iin kullanacaęımız teknik ne olabilir? Tek kelime ile: T6mevarım. Bu her zaman iyi bir yanıttır, evet. T6mevarım size "ben buradayım" diye seslenecektir 6nk6 bunun  $n=2$  durumunda dıřb6keylięin tanımı kapsamında doęru olduęunu biliyoruz. Taban durumumuz da olduka net. Aslında burada daha da basit bir taban durumu var; bu durumda  $n=1$  'dir. Eęer  $n=1$  ise, elinizdeki bire efit olacak Őekilde toplanacak tek sayı olduęundan ,  $\alpha(1) = 1$  olur. B6ylece burada herhangi bir Őey olmaz. Bu bařka bir yaklařımla (1kere  $x_1$ ) en ok 1kere  $f(x_1)$  olur demektir; ve bu da efitlilik anlamında ok heyecan verici bir Őey deęil. Yani bu durumda aslında  $n=2$  taban durumuna bile ihtiyacımız yok. Őimdi ilgin olan kısım, aslında daha tam anlamıyla ilgin de deęil ama, bizim t6mevarım adımıdır. Bu t6mevarım teknięinde iyi bir pratik olacak. Őimdi bizim ilgilendięimiz Őey doęrusal bileřimin  $f$  fonksiyonu, yani  $\alpha_k x_k$  ilgin birleřiminin t6m  $k$  deęerleri 6zerindeki toplamının fonksiyonudur. Őimdi t6mevarımı uyguluyoruz. T6mevarımsal olarak bildięim; diyelim ki Őu  $f$  toplamı  $n$ ' ye kadar gitmek yerine  $(n-1)$ 'e kadar toplanıyor. Bundan daha k66k her toplamla iři t6mevarımla 6zebiliriz. Bu durumda yapacaęımız Őey  $n$ ' ninci terimden kurtulmaya alıřmaktır. Bu deęeri ayırmak istiyorum. Ve tabii daha 6nce ilgin birleřimlerle uęrařtıysanız bu da ok doęaldır. Ama biraz cebir gerektirir. Yani bu  $\alpha_k x_k$  terimini buradan ayırmak istiyorum. Ve onun dıřında da bunu bir ilgin birleřim yapmak istiyorum. İřte numaramız burada. Af edersiniz burada  $f$  olmayacak. Eęer bu son terimi ıkarırsam  $\alpha_k$  'ların  $1$ 'den  $(n-1)$ 'e kadar olan toplamları  $1$ 'e efit olamaz; daha k66k bir Őeye efit olabilir. Bu nedenle bu terimi hemencecik ıkaramam. Burada bir numara yapmam lazım;  $x_k$  gibi. İyi.

Bunun neden doęru olduęunu g6rebilirsiniz, 6nk6  $(1-\alpha_n)$  ler birbirini g6t6r6r. Ve b6ylece ben sadece  $(\alpha_k x_k)$  elde ederim; tabii  $k$ 'nın  $1$ 'den  $(n-1)$ 'e kadar artan deęerleri iin. Artı  $\alpha_n x_n$  terimi olur. Őimdi ben burada herhangi bir Őey yapmadım. Bunlar birbirine efit. Ama Őimdi elimde garip bir 6zellik var: Bir tarafta bu iki sayı yani  $\alpha_n$  ve  $(1-\alpha_n)$  bire efit oluyor. Ve 6teki tarafta da, her Őeyi doęru yaptıysam, bu sayıların

toplamı da 1'den (n-1)'e kadar bire eşit oluyor. Bunların toplamı neden 1'e eşit oluyor? Çünkü bu sayıların toplamı  $1 - \alpha_n$ 'e eşit çıkmıştı. İki tarafı da  $1 - \alpha_n$ 'e bölersem; bunların toplamı 1 olur. Şimdi elimde iki tane ilgin birleşim var. Şimdi bildiğim iki şeyi uygulayacağım. Bildiğim bu ilgin birleşimin çalışacağıdır. Bir dakika, neden çalışacaktır? Bunun  $\alpha_n f(x_n) + (1 - \alpha_n)$  çarpı bu saçma f fonksiyonu olduğunu nasıl söyleyebiliyorum? Haydi, yanıt verin. Bunun iki olası yanıtı var. Bunlardan bir tanesi doğrudur ve bir tanesi yanlıştır. Pekiyi bunların hangisi olacak? Bunun küçük-eşit olması gerekiyordu. Bu önemlidir, bakın tahtada yazıyor. Çok zor olmamalı. Burada bunu bir büyük x değeri gibi değerlendiriyorum. Yani elimde bir  $x_n$  ve birde saçma bir X var. İstedğim şey bu iki büyük X değerlerinin ilgin birleşiminin en çok şu x değerlerinin fonksiyonlarının ilgin bileşimlerine eşit olması. Öyle mi? Bu aslında n'nin ikiye eşit olması durumundaki tümevarım hipotezidir diyorsunuz ama biz maalesef n=2 durumunu özel bir taban durumu olarak kanıtlamadık. Bu nedenle burada tümevarımı taban durumu ile ilgili olarak kullanamam. Eğer siz n'nin ikiye eşit olması taban durumunu kanıtladıysanız, bunu yapabilirsiniz. Ama burada bunu yapamayız. Bu durumda diğer yanıt dışbükeylik, doğru cevap.

Ve o da işte burada. Yani f dışbükeydir. Bunun doğru olduğunu, herhangi iki x değeri için geçerli olduğunu biliyoruz; tabii bu ikisinin toplamının 1 olması durumunda. Yani bunun doğru olduğunu biliyoruz. Tabii bu tümevarımı uyguladığımız durumda oluyor. Şimdi bu sağdaki terimle tümevarım yoluyla oynayacağız. Dikkat edin, daha önceden n'nin 2'den daha büyük bir tam sayı olabileceğini bilmiyorduk. Ama n'nin (n-1)'den daha büyük olduğunu biliyoruz. Bu kadarından emin olabilirim. Böylece bu  $(1 - \alpha_n)$  kere k'nin 1'den (n-1)'e kadar olan artışında,  $[\alpha_k / (1 - \alpha_n) \text{ kere } f(x_k)]$  'nın toplamıdır. Eğer bunu doğru yazdıysam; tümevarım hipotezi yoluyla.. Bu  $\alpha_k$ 'lar bölü,  $(\alpha_k / (1 - \alpha_n))$  toplamının bire eşit olduğunu biliyoruz. Burada bu  $(1 - \alpha_n)$ 'ler birbirini götürür ve istediğimizi elde ederiz. Bu k'nin 1'den n'ye kadar olan artımında  $\alpha_k f(x_k)$  toplamına eşittir. Böylece toplamın f'sinin en çok toplamların f'lerine eşit olduğu sonucunu elde ederiz. Bu da bizim önkuramımızı kanıtlamış olur. Evet, biraz dağınık oldu ama her adım kendi içinde oldukça açıktı. Katılıyor musunuz? Bu noktadan sonra artık Jensen'in eşitsizliğini kanıtlamak oldukça kolay olacak. İşte sihir buradaydı; ondan sonra da beklenenin çözümlenmesini yapmaya başlayacağız. Burada da bizim iyi dostlarımız olan göstergesel rastgele değişkenleri kullanacağız. Ama ondan önce şimdi bu söylemin doğruluğunu kanıtlamak zorundayız. Eğer elimizde bir dışbükey fonksiyon varsa beklenenin f'si en çok şu rastgele değişkenin f'sinin beklenenine eşittir. Tamam, bu bir rastgele değişken değil mi? Eğer bu rastgele değişkenden örnekleme yapmak isterseniz büyük X'den örnek alırsınız ve f'yi buna uygularsınız. Bu f(X) simgeleminin anlamı budur, çünkü X bir rastgele değişkendir. Burada f'nin dışbükey olması özelliğini kullanacağız. Bu aslında, umulanın tanımını hatırlarsanız, o kadar zor değildir. Bir dakika, burada bir varsayım daha yapmak istiyorum ve bu varsayım da X'in entegral yani tümlenik olduğudur; X bir tam sayı rastgele değişkendir. Yani tam sayı değerlerini aldığı anlamına gelir. Şimdi bizim bütün ilgilendiğimiz bu çünkü biz koşma sürelerine bakıyoruz. Bu anlatım aslında sürekli rastgele değişkenler için de doğrudur ama ben burada discrete yani

ayrık durumlara bakmak istiyorum, çünkü bu durumda  $E[f(X)]$ ' i yazabilirim. Evet,  $E[f(X)]$ 'in tanımı nedir?  $X$  sadece tamsayı değerleri alabiliyor. Bu oldukça kolay ama bunu hep aklınızda tutmanız gerekir. Aslında bu iyi bir alıştırmaya olacak.  $X$  hakkında çok fazla şey bilmiyorum sadece tamsayı değerleri aldığını biliyorum.  $X$ 'in beklenenini nasıl genişletebileceğim konusunda önerisi olan var mı? Bunu ezberleyen kaç kişi var? O zaman işimiz kolay olmayacak. Aslında beklenenin olasılıklarla ilgisi var, değil mi? Bu durumda bakmam gereken büyük  $X$ 'in olasılığının bir  $x$  değerine eşit olması durumu olabilir. Bu yapılacak iyi bir şey gibi görünüyor. Buraya başka ne gelebilir? Bir toplam, evet bu toplamda  $x$  eksi sonsuzla artı sonsuz arasında herhangi bir yerde olabilir. Bu gerçekten de doğrudur. Ve elimizde biraz daha bir şeyler var. Burada eksik olan bir şey var. Bu toplam nedir? Bu toplam rastgele değişken olan  $x$ 'in tamsayı değerleri alması durumunda hangi sonucu verir? 1 verir, güzel. Bu durumda buraya bir şey daha eklemem lazım, yani  $x$ 'i eklemem lazım. Evet beklenenin tanımı işte budur. Şimdi bir şeylerin toplamı olan  $f$ , buradaki katsayıların toplamı 1'e eşitse, biraz önce kanıtladığımız önkurama çok benzer olacak. Biz bunu sonsuz olmayan durumda kanıtladık ama sonuç öyle çıkar ki, tüm tamsayıları alsanız bile bu doğru olur. Şimdilik bunu sadece varsayacağım. Elimde bu olasılıklar ve toplamları 1'e eşit olan şu  $\alpha$  değerleri var. Bu nedenle şu eşitsizliği kullanabilirim ve bu en çok  $x$ 'in eksi sonsuzla artı sonsuz arasında değişen değerlerinde yapılacak toplamalarım var; bu işlem altındaki alfalar büyük  $X$ 'in küçük  $x$ 'e eşit olma olasılıkları çarpı  $f(x)$  olur. Evet işte budur. Ben buraya kadar önkuramı kullandım; şimdi belki önkuramı söyleyebilirim. Ama dikkat ederseniz burada bir hile yaptım; aslında önkuramın sayılabilir versiyonunu kullandım; bu durumda sadece sonsuz olmayan durumu kanıtlayabilirim.

Ama bu derste yapabileceğimin en fazlası bu olabilir. Bu önkuramımız. Şimdi burada kanıtlamak istediğim şey bu toplamın en çok  $E[f(x)]$  olduğudur. Aslında buna eşittir. Baktığınız zaman da eşit gibi görünüyor, değil mi? Elinizde bir takım olasılıkların toplamı kere  $f(x)$  var. Bu aslında  $f(x)$  in tanımı gibi görünüyor ama öyle değil. Burada biraz dikkatli olmanız lazım çünkü  $E[f(x)]$ 'in aslında  $f(x)$  in belirli bir değere eşit olma olasılığını göstermesi gerekir. Bu ikisini şöyle ilişkilendirebiliriz. Çok zor bir şey değil.  $f$ 'in aldığı tüm değerlere bakabilirsiniz ve ondan sonra da  $x$ 'e eşitlenebilecek tüm  $k$  değerlerine bakabilirsiniz. Yani  $f(k)$ 'nin  $X$ 'e eşit olduğu bütün  $k$  değerlerinin toplamında  $x$ 'in  $k$ 'ya eşit olduğu olasılığı ele alabilirsiniz. Bu  $f(X)$ 'in  $x$ 'e eşit olma olasılığının başka bir anlatım biçimidir. Böylece, başka bir anlatımla terimleri özel bir şekilde grupluyorum. Diyorum ki  $f(X)$  bazı değerler alıyor. Burada bir düzeltme yapmam akıllıca olur; daha önce  $k$ 'ları tanımsız kullanmıştım. Bu nedenle şunlara başka bir şey diyeyim. Burada bunlara  $y$  diyeyim. Af ederseniz, burada simgelemi değiştiriyorum. Ama böyle daha anlamlı oluyor.  $X$ 'in  $x$ 'e eşit olması olasılığına bakmalıyım. Burada beni ilgilendiren bu  $f(X)$  değerlerinin hangi değerleri aldığıdır. Buna  $y$  diyelim ve  $y$ 'nin  $f$  olabilecek hangi değerleri aldığına bakalım; bu  $f$ 'in aralığıdır. Sonra  $x$ 'in bütün değerlerine  $f(X)$ 'in  $y$ 'ye eşit olması durumunda bakacağım. Bu olasılıkların hepsini toplayacağım, çünkü burada  $x$ 'in değişik değerleri olduğunda bunlar bağımsız olaylardır. Böylece  $f(X)$ 'in  $y$ 'ye eşit olma olasılığı bu toplam olacak. Buradaki büyük  $X$ 'dir. Bu küçük  $y$ 'dir. Ve sonra bunu  $y$  ile çarparsam, o zaman  $f(X)$ 'in

beklenenini elde ederim. Şimdi bu konuda düşündüğünüzde bu iki eşitsizliğin geçerli olduğunu görürsünüz. Bu burada biraz garip görünebilir, çünkü şu toplamlar potansiyel olarak sonsuz olabilir. Ama doğrudur. Ve böylece Jensen'in eşitsizliği kanıtlanmış olur. Gördüğümüz gibi o kadar zor değildi, birkaç tahta tuttu, ama temelinde bu güçlü dışbükeylik önkuramı vardı. Yani burada sadece dışbükeyliği kullandık. Ve aynı zamanda  $E(X)$  in tanımını kullandık. Dışbükeyliği kullandık ve bu bize küçük  $f$ 'leri içeri alma şansı verdi. Sonra bu terimleri tekrar grupladığımızda gördük ki çıkan  $E[f(x)]$  miş. Yani buradaki tek eşitsizlik dışbükeylikten kaynaklandı.

Evet, şimdi sırada algoritmalar var. Şu ana kadar sadece temel olasılık olgularıyla uğraştık ve bu bize iyi bir pratik sağladı. Bununla testlerde karşılaşabilirsiniz onun için şaşırmayın. Bu benim içinde bir çözülmesi gereken durumdu. Algoritmalar söz konusu olunca sezgiye çok ihtiyacınız olur. Her hangi bir şey algoritma ile ilgili olunca bu yaklaşım çok akılcıdır, çünkü bildiğiniz gibi bazı şeylere dayanmak durumundasınız, çünkü siz bilgisayar bilimcilerisinizdir veya o türden bir şeysinizdir. Bu dersin kapsamında siz bilgisayar bilimcilerisiniz. Ama burada olduğu gibi temel olasılıklar için içindeyse ve siz de matematikçi değilseniz, bu daha az sezgisel bakış yaratır ve anlaması o kadar kolay değildir. Ve testlerde hız oldukça önemlidir. Final sınavınızda da hız önemli olacaktır. Ev ödevleri size kesinlikle zarar vermez; ev ödevi sınav tarzı enteresandır, çünkü akıllı olmayı gerektirir. Aslında yaratıcı olmak zorundasınız ve bu algoritma tasarımını gerçekten sınavan bir durumdur. Şu ana kadar sadece çözümlere yettimizi sınavdık ve olasılıklarla ne kadar çalışabileceğimizi gördük yani örneğin rastgele çabuk sıralamanın yürütüm süresi neydi, hatırlıyor musunuz gibi şeylerle uğraştık. İkinci test aslında yaratıcılığınızı sınavacak, çünkü daha fazla zamanınız olacak. İki saatte yaratıcı olmanız biraz zordur. Evet, sonuçta rastgele yapılandırılmış bir ikili arama ağacının beklenen yüksekliğini çözmek istiyoruz.

Bunu daha önceden tanımlamıştım ama tekrar edeyim; bu bir sene önce bu dersin en başındaydı. Ben  $n$  düğümü olan rastgele yapılandırılmış bir ikili arama ağacının yüksekliği ile ilgili rastgele değişkeni alacağım. Yani buradaki  $n$  değerleri rastgele haldedir ve bir rastgele permütasyon kullanarak bu değerleri soldan sağa doğru ağacın içine yerleştireceğim. Ne yükseklikte bir ağaç elde edersiniz? Her düğümün maksimum derinliği nedir?  $X(n)$ 'e fazla bakmayacağım; bunun yerine  $X(n)$ 'in üstel haline bakacağım. Bu durumda hala bir şeyler sezemeyebiliriz, ama  $2$ 'nin  $X(n)$  üsteli bir dışbükey fonksiyondur ve şöyle görünür: Bu oldukça diktir. Bu yapabildiğim en iyi çizim; histogramı nasıl çizdiğimi görmüştünüz. Bundan sonra bu rastgele değişkeni cebir ile bir şeyler olarak yazmak istiyorum. Bu yöntemde ana uygulama bunu durumlara bölmektir. Bu bizim takip ettiğimiz genel yoldur, çünkü burada olup biteni anlatan çok fazla senaryo vardır. Yani bir ağacı en baştan nasıl yapılandırmaya başlarız? İlk yaptığımız şey birinci düğümü ele almaktır. Bunu çizimimize koyarız ve kök yaparız. Yani dizilimimizdeki ilk değeri, ki daha önceden bunun sıralamada nereye geleceğini bilmeye imkan yoktur, kök olarak tanımlarız. Ve bu kök olarak kalır. O noktadan sonra hiçbir şekilde kökü değiştirmeyiz. Bu evreden sonra kalan

elemanların bir kısmı kök değerinden daha küçüktür onun için onlar buraya gelir. Şimdilik kökteki bu değere  $r$  diyelim. Ve bunların bazıları da  $r$ 'den büyüktür. Onlar da şuraya gider. Belki burada daha fazla eleman vardır, belki de şurada. Kim bilir? Rastgele bölüntüleme daha doğrusu tekbiçimli rastgele bölüntüleme çerçevesinde, ki bunun size tanıdık gelmesi gerekiyor, burada  $k$  sayıda elemanın olması ve şurada da  $(n-k)$  sayıda elemanın olmasının olasılığı, herhangi bir  $k$  değeri için eşittir; çünkü bu seçim tekbiçimli yapılmıştır. Kök tekbiçimli olarak seçilmiştir. Rastgele bir permütasyonun içindeki ilk elemandır. Bu nedenle bunu kullanarak parametreler oluştururum. Burada kaç eleman var ve şurada kaç eleman var? Bu bir rastgele yapılmış ikili arama ağacı olduğundan, şurada kaç boğum olacağı belirlenmiştir; çünkü kimin solda kimin sağda yer alacağı  $r$  seçimine bağlıdır. Ve böylece çabuk sıralamadaki gibi bölüntülemelerimi yapabilirim.  $r$  nin solunda ve sağında kalan elemanları kullanarak, özyineleme ile rastgele yapılandırılmış bir ikili arama ağacını iki alt permütasyon olarak kurularım. Çünkü tekbiçimli permütasyonların alt permütasyonları da tek biçimlidir. Yani sonuçta bunlar özyinelemeli problemlerdir. Ve biz özyinelemeli problemlerin nasıl çözümleneceğini biliyoruz. Bütün bilmemiz gereken burada  $(k-1)$  elemanının olduğu ve şurada da  $(n-k)$  elemanının olduğu. Ve bu  $r$  nin  $k$  rütbesinde olduğu anlamına gelir; hatırlarsanız rütbeyi ben sıralanmış dizilimlerin dizini bağlamında kullanıyorum. Şimdi nereyi kullanmalıyım? Eğer kök  $r$ 'nin rütbesi  $k$  ise ve bu da rastgele olan bu olayın durumu ile ilgili bir veri ise, o zaman  $X_n = 1 + \max.[X_{k-1} X_{n-k}]$  olur. Çünkü bu ağacın yüksekliği her iki alt ağacın maksimum yüksekliklerinin toplamı artı 1'dir; çünkü en yukarıda bir düzeyimiz daha var. Yani yapılması gereken doğal şey budur. Ama çözümlenmeye çalıştığımız  $Y_n$ . Bu nedenle  $Y_n$ 'yi 2'nin bir kuvveti cinsinden yazmamız gerekiyor. Böylece bu iki kere 2'nin  $\max.(X_{k-1})$  inci kuvveti, ki bu  $Y_{k-1}$ 'dir kere 2'nin şu değerli kuvveti, ki bu da  $Y_{n-k}$ 'dir. Şimdi burada neden  $X$ 'lerin yerine  $Y$ 'lerle ilgilendiğimizi görmeye başlayabilirsiniz; çünkü bildiğimiz şeylerle uğraşmak isteriz. Biz özyinelemeyi çözmeye çalıştığımızda, örneğin beklenen koşma süresi ile ilgili bir şeyi çözmeye çalıştığımızda ki daha burada bekleneni hesaplamadık; ama çabuk sıralamanın beklenen koşma süresini hesaplamaya kalktığımızda biliyoruz ki elimizde iki kere bir şey olur yani birden fazla özyinelemeli alt problem vardır ve bundan sonra birbiri ile toplanır.

Burada bir adet iki faktörü var Burada da bir maksimum var. Ama sezgisel olarak biz rastgele değişkenleri bir sabitle nasıl çarpacağımızı biliyoruz çünkü burada iki tane birbirine yaklaşık eşit özyinelemeli alt problem var; bunlar ne olduklarını bilmesek de şu ikisinin maksimumu kadar. Ama burada artı 1 var ve bunu nasıl kullanacağımızı henüz bilmiyoruz. Diğer taraftan bizim kullandığımız teknikler özyinelemeli çözümlerin sabit faktörlere kadar olan durumlarında oldukça iyiler. Öte yandan bu artı 1 değeri bizim sabit faktörümüzü çok fazla etkilemeyecek gibi görünüyor. Ama bu önemli bir sorun olabilir çünkü üstel uygulamalarda bu 2 faktörüne yol açar. O nedenle burada şu artı 1 değerinin ne olduğunu görmek o kadar kolay değildir. Şimdi benim burada  $X_n$  ile yapacağımı gözlemler ve bunu evde denerseniz bu artı 1 değerinin

kendiliğinden yok olacağını ve size bir sınırlama getirmeyeceğini görürsünüz. Bunu sadece kanıtlayamazsınız. Ama iki faktörü ile oldukça iyi durumda oluruz. Çünkü bununla nasıl başa çıkabileceğimizi biliyoruz.

Bu kanıtlamayı bitirince neden  $X_n$  yerine  $Y_n$  kullandığımızla ilgili söyleyeceğimiz başka şeyler de olacak. Ama şimdilik sadece  $Y_n$ 'yi kullanacağız. Böylece bu bir tür özyineleme ama sadece bu olaya uyarlanmış. Pekiyi, bunu her zaman geçerli olan bir anlatım haline nasıl getirebilirim? Pardon? Olayın olasılığına bölerek mi? Hem evet, hem hayır. Doğru, bu iki olay birbirinden bağımsız. Ya da daha iyi bir anlatımla olasılıkları birbirine yakın. Tamamıyla birbirinden bağımsız değil. Aslında bir tanesi ötekileri belirliyor. Bu durumda bu tip bir olguyu cebirle nasıl tanımlayabilir, nasıl ifade edebilirim? Göstergesel rastgele değişkenlerle, güzel.

Eski arkadaşlarınız göstergesel rastgele değişkenleri hatırlıyorsunuz değil mi? Bütün bu analizler hep göstergesel rastgele değişkenleri kullanır. Bu olayı da onlarla anlatacağız ve buna  $Z_{nk}$  adını vereceğiz; değeri eğer kökün rütbesi  $k$  ise 1 olacak diğer durumlarda 0 olacak. Böylece özellikle bütün  $k$  değerleri denendiğinde  $n$ 'nin belirli bir değerinde bunların olasılıkları eşit olacaktır. Bunun 1'e eşit olma olasılığı aynı zamanda göstergesel rastgele değişkenin de beklenendir ki bunu biliyor olmanız lazım -- beklenen ya 1 yada 0 olabilir. Beklenende 0'ın bir etkisi olmaz; bu nedenle sonucun  $1/n$  olacağını sanıyorum; tabii eğer doğru işlem yaptıysam.

Bu durumda kökün rütbesinin ne olacağı konusunda  $n$  sayıda olasılık vardır. Bu tek biçimli bir permütasyon olduğundan bunların her birinin olasılığı eşittir. Böylece şimdi bu şartlı anlatımı bir toplam olarak yazabilirim ve bu toplamda  $Z_{nk}$ 'lar hangi durumda olduğumu seçmeme izin verirler. Şimdi  $Y_n$  eşittir toplam  $k$ , 1'den  $n$ 'ye kadar  $Z_{nk}$  (2 kere  $Y_{k-1}$  ile  $Y_{n-k}$ 'nin çarpımlarının maksimumu) değerlerini alır. Şimdi eski arkadaşımız yineleme ile karşılaştık ve bunu çözmemiz gerekiyor. Ama bunu şu evrede çözemeyiz, çünkü bu bir rastgele değişkendir ve özyinelemeli rastgele değişkenlerden bahsediyoruz. Onun için önce her iki tarafın da beklenenini ele almamız lazım. Sınırlayabileceğimiz tek şey de bu olabilir.

$Y_n$  şanssız bir durumda  $n^2$  olacaktır, af edersiniz  $n^2$  olmayacaktır; aslında  $n^2$  de olabilir. Ama eğer şanssızsak  $2^n$ 'ye doğru da gidebilir, çünkü  $X_n$ ,  $n$  boyutunda, yani ağacın boyu kadar olabilir. Ve  $Y_n$ 'de  $2$ 'nin bu değerdeki kuvvetidir. Yani  $2^n$ 'ye ulaşabilir. Burada kanıtlamak istediğimiz bu değer  $n$  cinsinden polinomsal olduğudur. Eğer bu değer  $n$ 'nin belli bir sabit üzerindeki kuvvetinde ise ve burada logaritma alırsak o zaman bunun düzeyi logaritma  $n$  düzeyi olur. Bu nedenle bekleneni ele alacağız ve sonucun bunu garanti etmesini umacağız. Evet, elimizde rastgele değişkenlerin toplamının bu bekleneni çarpı özyinelemeli rastgele değişkenler var. Hay Allah burada bir köşeli parantezi unutmuşum. Şimdi bu çözümlenmede yapmamız gereken ilk şey nedir?

Evet, beklenenin doğrusallığı; bunu hatırlamak kolaydı. Şimdi bir toplamımız var. E' yi bunun içine koyalım. Evet şimdi çarpımımızın beklenenini elde ettik. Burada ne kullanmalıyız? Bağımsızlık prensibini. Bunların birbirinden bağımsız olduğunu umuyoruz. O zaman bunu yazabiliriz. Eğer öyleyse bu da çarpımın bekleneni olacaktır. Burada şu 2'yi de köşeli parantezin dışına çıkaralım, çünkü onu burada tutmanın hiçbir yararı yok.

Buradaki Y'ler X gibi görünmeye başladı; ben bile artık bunları okuyamıyorum. Bunun için özür dilerim. Bunların hepsi Y olmalıydı. Ve bunların hepsi akıllı rastgele değişkenler. Pekiyi, bunlar neden bağımsız? Burada baktığımız şey kökün seçimi ile ilgili; kökün rütbesinin n boyutunda bir problemde ne olduğuna bakıyoruz. Burada ise köke bakıyoruz ama esas baktığımız bu tip arama ağaçlarında kökün sol tarafında neler olduğu ve sağ tarafında neler olduğudur. Bunlar bağımsız seçimlerdir. Çünkü burada her şey tekbiçimlidir. Onun için bu elemanın seçimi tekbiçimli oldu ve bu da sol ve sağ taraflarla ilgili bölüntüleri sağladı. Sol ve sağdaki alt ağaçların kökünün kim oldukları tamamen bağımsız özyinelemeli seçimlerle belirlendi. Bu nedenle bu alıştığımızdan biraz daha karmaşık bir durumdur.

Daha önce bu algoritmalar da rastgele seçimlerdi; şimdi ise bunlar başlamadan önce seçtiğimiz rastgele sayılar ve yapılandırılmaya sonradan gidiliyor. Bu komik bir durum ama bunlar hala birbirinden bağımsız ve bu nedenle biz bunları aynı çabuk sıralamada olduğu gibi elde ediyoruz. Tamam. Şimdi devam edelim. Ve artık biraz dağınık olmanın zamanı geldi.

Burada bildiğimiz bir şey var.  $E[Z_{nk}]$ 'yi buraya yazmıştık. Bu  $1/n$  idi. Bu güzel bir değer. Böylece dışarıda  $2/n$  elde ediyoruz ve bu iki şeyin beklenenlerinin maksimum toplamını da burada elde ediyoruz. Normal olarak bunu T'nin maksı veya y'nin maksı olarak yazarız. Ama burada yazmanız gereken bu 2 değişkenin en büyük değeri olduğu şeklindedir.

Ve buradaki kandırmaca da aslında çok büyük bir numarada değildir ama bu maks'ın en çok toplama eşit olduğudur. Hatırlayın burada negatif olmayan değerler vardı. Böylece  $2/n$  çarpı k'nın 1 den n'ye kadar olan değişiminin maks yerine toplamların bekleneni elde edilir. Şimdi bu evre bir bakıma bizim sınır tanımlamamızda bir şeyleri kaybettiğimiz adımdır. Şu ana kadar hep kesin anlamlı gitmiştik. Şimdi bayağı bir dağınık çalışıyoruz. Bu maks'ın en çok toplam olduğu doğrudur. Ama işler ilerledikçe bu bayağı gevşek bir üst sınır hale gelir. Bunu ilerisi için aklımızda tutmak zorundayız. Bu toplamayla başka ne yapabiliriz? Şimdi bunun size tanıdık geleceğini umuyorum.

Şimdi elimizde iki şeyin toplamı varken ben bunu tek şeyin toplamı halinde değerlendirmek istiyorum. Pardon? Evet, beklenenin doğrusallığını kullanabilirsiniz. İyi. Evet, bu yapmam gereken ilk işlem. Beklenenin doğrusallığı kavramı benim bunu ayırmama izin verir. Şimdi elimde  $2n$  şeyinin toplamı var. Ve ben bunu şu elemanların toplamı ve bu elemanların toplamı olarak ikiye bölmek istiyorum. Bu iki toplam hakkında bildiğimiz bir şey var mı? Bu iki toplamla ilgili bilinen bildiğimiz bir şey var



mı? Bunların ikisi de aynı. Buradaki terimlerin her biri ikişer defa görünüyor. Birincisi  $k-1$  diyor. Ötekide  $n-k$  diyor. Ve bunlar tek sayı bile olsa bu yine geçerli olacak. Yani bu durumda bu toplamlardan sadece bir tanesini alıp sonra bunu 2 ile çarpabiliriz. Bu durumda bu  $4/n$  çarpı şu toplam olur ve ben bunu  $k$ 'nın 0'dan  $n-1$ 'e kadar olan değerlerindeki toplam olarak yazacağım; bu da  $E[Y_k]$  olacak.

Şimdi kontrol etmeniz gereken  $[Y_k]'$  nın 0 dan  $n-1$  e kadar olan değişimlerde 2'şer defa ortaya çıkmasıdır. Yani benim şimdi bir özyinelemem var. Şuradaki  $E[Y_n]$  en fazla bu değer olabilir. Bunu hafızada tutmak için bir yere yazalım. Şimdi bu nasıl oldu? İyi oldu. Şimdi bu yinelemeyi çözmek durumundayım. Böyle karmaşık bir yinelemeyi nasıl çözebilirim? Yerine koyma metoduyla. Evet. Ana metotla değil. Kabul ediyorum bu oldukça karmaşık bir yineleme. Bu nedenle bir tahmin yapacağım. Ve bu tahminimi zaten size söylemiştim. Bu  $n^3$  olacak. Bana göre  $n^3$  bu kanıtın elde edilebileceği en doğru başlangıç noktası.

Yerine koyma metodu tümevarım yoluyla yapılan bir kanıtlama metodudur. Ve her tümevarımla kanıtlama yönteminin iki yönü vardır; aslında çok özentili değilse hemen her kanıtlamanın iki yönü vardır. Önce bunun bir taban durumu olmalıdır. Ve buradaki taban durumu  $n = O(1)$ , yani birinci düzeyde olma durumudur. Bunu yazmadım ama tabii ki eğer sabit bir ağaç boyunuz varsa bunun yüksekliği de sabittir. Bu durumda bunun doğru olması için bizim  $c$  yi yeterince büyük seçmemiz gerekir. Yani bunu hiçbir zaman unutmamalısınız. Birçok kişi bunu testlerde unuttur. Burada biz taban durumunu bile belirttik ama genelde taban durumunu fazla belirtmeyiz. Ve burada mutlaka bir tane olduğunu varsaymanız gerekir. Ve tümevarım yöntemiyle yapacağınız her kanıtta da bunu belirtmeniz gerekir. Şimdi tümevarım adımındayız. Burada iddiam  $Y_n$ 'nin bekleneninin en çok  $c$  çarpı  $n^3$  olduğu ve küçük değerli  $n$ 'ler için bunun geçerli olduğudur. Bu evrede siz tümevarım hipotezini yazmalısınız ama ben bunu yapmayacağım çünkü zamanım giderek kısalıyor. Şimdi elimizde bir  $E[Y_n]$  var ve bu en çok da şu değere eşit. Yani  $E[Y_n]$  en çok  $4/n$  ve  $k$ 'nın 0'dan  $(n-1)$ 'e kadar değişen  $Y_k$  değerlerinin beklenenlerinin toplamına eşittir.

Şimdi dikkat ederseniz  $k$  her zaman  $n$ 'den daha küçüktür. Bu durumda biz tümevarımı uygulayabiliriz. Şimdi bu en çok  $4/n$  çarpı  $k$ 'nın 0'dan  $(n-1)$ 'e kadar değiştiğinde  $c$  kere  $k$  küpün toplamına eşittir. Bu tümevarım hipotezidir. Hoş bir durum. Şimdi bu toplam için bir üst sınıra ihtiyacım var. Ve eğer hafızanız iyiyse bu toplamın kapalı formunu anımsayabilirsiniz. Ama benim eskiden olduğu kadar iyi bir belleğim yok. Çocukken bu toplamı hiçbir zaman ezberlememişim. Ve 12 yaşımdan beri ezberlediğim her şeyi de hatırlayamıyorum.  $\pi$  'nin her ondalık sayısı gibi şeyleri hatırlıyorum. Ama şimdi ezberlemeye çalıştığım herhangi bir şey aynı şekilde kalıcı olmuyor. Bu nedenle bu toplamın ne olduğunu bilmiyorum.

Bu toplamı yaklaşık bir değere getirecek iyi yol ne olabilir? Entegral, evet, iyi. Bu nedenle  $c$  'yi dışarıya çıkaracağım. Böylece bu  $4c/n$  olur. Ve toplamda en çok bu entegral olur. Eğer aralığı doğru saptarsanız üst sınırı bir arttırabilirsiniz. Böyle bir

durumda  $n-1$  yerine  $n$  değerine çıkabilirsiniz. Bu kitapta var. Bu işlem monoton bir fonksiyonunuz varsa oldukça da sezgiseldir. İşte işin anahtarı budur. Bu nedenle elinizde şuna benzer bir şey olur. Ve bildiğiniz gibi toplam bunların her birini almak ve bunları 1 değeri ile ağırlandırmakla yapılır. Buradaki integral bu eğrinin altındaki alanı hesaplamak anlamına gelir. Özellikle bu durumda integralin yaklaşılaştırılmasıyla ilgili söyleyeceğim şey, bu işlemin sonuna kadar birer değer arttırılarak gidildiğinde toplam sonunda bu integral olacaktır. Şimdi bu şekilsel bir kanıtlama oldu. Ama bunu kitapta okuyabilirsiniz. Bunu da daha önceki derslerinizden birinden bildiğinizi sanıyorum.

Şimdi umuyorum ki entegralleri çözebilirsiniz.  $x^3$  ün integrali  $x^4/4$  dür. Sonra bunu  $n$  sayısında değerlendireceğiz. Ve bu 0 olacak. Buradan 0 çıkarmak o kadar önemli değil çünkü 0'ın 4. kuvveti de 0' dır. Böylece bu  $n^4/4$  oldu. Yani  $4c/n$  çarpı  $n^4/4$  'e eşit olur. Ve çok uygun bir şekilde bu 4 öteki 4 ü götürür. Ayrıca bu 4, 3' e dönüşür ve elimizde  $n^3$  kalır. Sonuç  $c n^3$  'dür. Bu bizim için çok uygun çünkü zaten kanıtlamak istediğimiz buydu. Evet, bu sonuç aslında hiç kalan terim olmadan ortaya çıktı. Aslında işi yaparken oldukça özensiz davrandık ama sonuç gerçekten şanslı olduğumuzu gösterdi. Yani burada doğru yerlerde özensiz davrandık demektir. Evet, gördüğünüz gibi bu çok şaşırtsmacalı bir kanıtlamadır. Bunu elle yapmaya kalksaydınız buradaki dağınıklıktan dolayı doğru yanıtı ulaşamama ihtimali vardı. Ama bu tam yerinde çalışır halde ortaya çıktı.

Şimdi geri kala bir dakikamda bununla ilgili bir şeyler söylemek istiyorum. Sonuca nasıl ulaştığımızı hatırlamaya çalışalım. Bunu yazmayacağım, çünkü yeterince zamanım yok. Biraz önce  $Y_n$  üzerindeki bir sınırı kanıtladık ve bu 2 nin  $X_n$  'inci kuvvetine eşitti. Bizim ilgilendiğimiz aslında  $X_n$  di. Bu nedenle Jensen'in eşitsizliğini kullandık. Burada 2 'nin  $E[X_n]$  'inci kuvvetinin en çok  $E[2^n X_n]$  kuvveti] değerine ulaşabileceğini gördük. Bunun böyle olduğunu biliyoruz çünkü şuradaki  $Y_n$ . Ayrıca biz  $E[Y_n]$  'nin  $n^3$  düzeyinde olduğunu biliyoruz. Bu sabiti taban durumu için yeterince büyük seçmek zorunda kaldık. Ve buradaki sabiti hesaplamak için de herhangi bir çaba göstermedik. Çünkü ondan sonra her iki tarafın da logaritmasını aldık. Ve bu durumda  $E[X_n]$  'nin en çok  $n^3$  düzeyinin logaritmasına eşit olduğunu bulduk. Bu sabit çarpımsal bir sabittir. Böylece logaritmalarınızı aldığınızda. artımsal bir yapıya bürünür. Bu sabit ise bir üsteldir. Bu nedenle logaritmasını alırsanız çarpımsal bir hale dönüşür.

$:3 \log n + O(1)$ . bu rastgele yapılanmış ikili bir arama ağacı için çok sıkı bir sınırdır. Yani ağacın beklenen yüksekliği ile ilgili sıkı bir sınırdır demeliyim. Aslında  $X_n$  'nin umulan yüksekliği yaklaşık olarak demeliyim, burada hassas olmak istemiyorum,  $2,9882 \log n$  dir. Bu benim arkadaşım Luke Devroy tarafından 1986 da bulunmuştur. O Montreal'deki McGill üniversitesinde bir profesördür. Yani biz buna bayağı yakın çıktık, çünkü 3 yerine 2,98 diyebilirdik. Ve bunu burada kanıtlamayacağım. Buradaki zor kısım aslında alt sınırdır. Ama bu da sadece bu kadardır.

Burada neden  $X_n$  yerine  $Y_n$  kullandığımızla ilgili birkaç şey daha söylemek istiyorum. Ve bu tamamıyla özensizlikle ilgilidir. Ve özellikle bu adımda şuradaki iki rastgele değişkenin maksimum değerinin en çok bu toplam olacağını söylemiştik. Ve bu hem X için hem de Y için doğrudur ama Y için daha da doğrudur. Tamam, bu biraz garipleşti. Çünkü hatırlayın, burada çözümlenmeye çalıştığımız şey k' nın bütün olası değerleri içindi. Bu k'nın değeri ne olursa olsun çalışmak zorundaydı. Yani bütün bu durumları aynı anda sınırlıyoruz ve hepsinin toplamlarını da sınırlıyoruz.

Yani burada biz aslında k-1'e karşı n-k'ya bakıyoruz. Ve gerçekte burada bir polinomsal versiyon var. Yani a ve b gibi iki değeri alıp a çarpı b nin en büyük değeri, en çok a+b olabilir dersiniz. Öteki taraftan şunu da diyebilirsiniz: 2 nin a'nıncı kuvvetinin maksimumu ve 2 nin b' ninci kuvvetinin maksimumu en çok  $2^a+2^b$  'ye eşittir. Bunu söylemek ötekenden iyi değil midir? Aslında her ikisi de tabii ki aynıdır. Ama öteki taraftan a-b' ye değerler büyüdükçe bakarsanız, bunun sıkı bir sınırlama oluşu şuna oranla daha hızlı olacaktır. Çünkü burada a-b arasındaki absolute yani mutlak farka bakıyoruz. Bu nedenle de bu oldukça iyidir ve bu da oldukça kötüdür. Üstelik bizim durumumuz eğer a ile b birbirlerine çok yakınsa daha da kötü olur.

Ama burada bunu k-1 ve n-k kapsayan bütün bölüntüleri çözmeye çalışıyoruz. Bu nedenle eğer ortada bazı durumlarda tam eşit bölündüklerinde sorunlar çıkarsa bu kabul edilebilir. Ama eğer bir skew yani kayıklık durumunda bu şuna çok yakın olacaktır. Ve şu da bundan çok uzak olacaktır. Burada çok fazla kayba uğramamak için kenara çok yaklaşmanız gerekir. Ya da şurada çok fazla kaybedersiniz. İşte bu sezgiseldir. Şimdi bunu deneyin ve  $X_n$ 'e ne olduğunu görün; olmayacağını görürsünüz.

Gelecek derste görüşmek üzere.