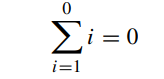
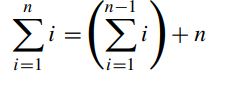
Bu durumda, sumFirstN fonksiyonunun karmaşıklığı O(1) olduğundan, bu fonksiyonu yazmak en iyisi olacaktır. Bu, bu fonksiyonun çalıştırılma süresinin, veri boyutu n'ye bağlı olmadığı anlamına gelir. Ancak, özyinelemeli bir fonksiyonu göstermek için toplama işleminin tanımına geri dönelim. Toplamın tanımının iki bölümü vardır. İlk olarak, tanımın temel durumu.

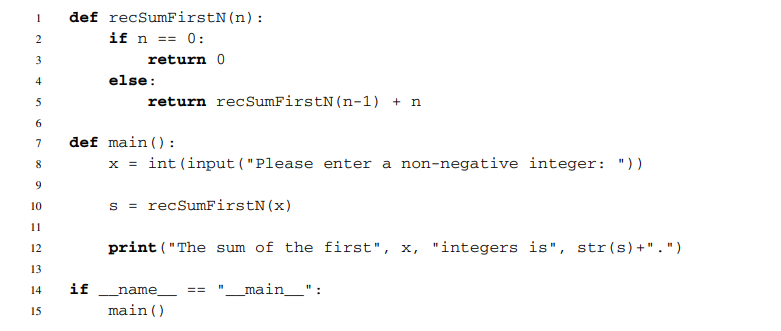


Tanımın özyinelemeli kısmı aşağıdaki gibidir. Buna bir özyinelemeli tanım diyoruz çünkü kendisiyle ilişkilendirilmiş bir şekilde tanımlanmıştır. Özyinelemeli tanımın dikkat çekici bir özelliği, bu durumda n - 1 olarak daha küçük bir n'ye dayalı olarak tanımlanmış olmasıdır. n - 1'e kadar olan toplam, özyinelemeli çağrımızdır ve işleyecektir. Eğer ilk 5 tam sayının toplamını hesaplamak istersek, özyinelemeli çağrı 1 + 2 + 3 + 4'ü hesaplayarak bize 10'u verir. n'yi eklediğimizde ise istediğimiz sonuç olan 15'i elde ederiz.



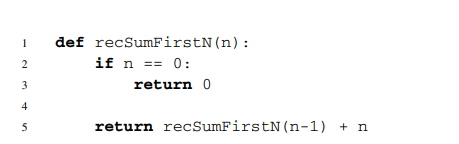
Bu özyinelemeli tanımın iki bölümü, doğrudan Python'da bir özyinelemeli fonksiyona çevrilebilir. Özyinelemeli tanım, Bölüm 3.4.2'de verilmiştir.

3.4.2 Tamsayıların Özyinelemeli Toplamı

Sect. 3.4.2'deki kodun içindeki recSumFirstN fonksiyonu özyinelemedir. Daha küçük bir değerle kendisini çağırır ve önce bir temel duruma sahiptir, bu nedenle iyi bir şekilde oluşturulmuştur.

Bu özyinelemeli fonksiyonda dikkat çekebileceğimiz bir şey var. else ifadesi gerekli değildir. Python yorumlayıcısı bir return ifadesiyle karşılaştığında, yorumlayıcı hemen döner ve fonksiyonun geri kalanını çalıştırmaz. Dolayısıyla, Sect. 3.4.2'de eğer fonksiyon if ifadesinin şu kısmında 0 döndürürse, fonksiyonun geri kalanı çalıştırılmaz. Eğer n sıfır değilse, o zaman else ifadesindeki kodu çalıştırmak isteriz. Bu, bu fonksiyonu Sect. 3.4.3'te gösterildiği gibi yeniden yazabileceğimiz anlamına gelir.

3.4.3 Else Gerekli Değil



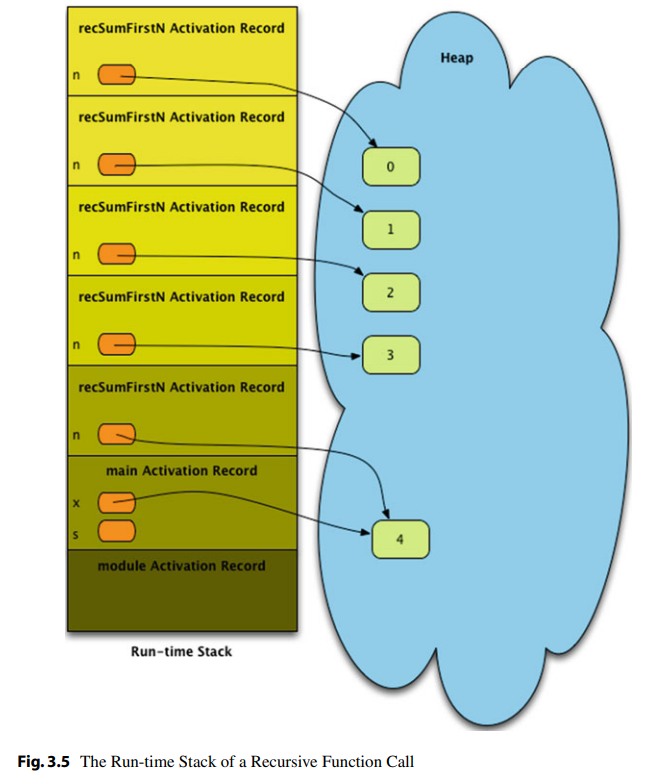
Sect. 3.4.3'teki kodun formatı, özyinelemeli fonksiyonları yazmanın yaygın bir yoludur. Bazı durumlarda özyinelemeli bir fonksiyonun birden fazla temel durumu olabilir. Her temel durum, içinde bir return ifadesi bulunan bir if ifadesi ile ele alınabilir. Eğer tüm temel durumlar bir return ile sonuçlanıyorsa, özyineleme durumu bir else içinde olmak zorunda değildir. Özyineleme durumu, özyinelemeli fonksiyon tanımında en sonda gelir.

3.5 Bir Özyinelemeli Fonksiyonun İzlenmesi

Bu bölümün başında, özyinelemeli bir fonksiyon yazarken "Çok fazla düşünme" direktifi verildi. Özyinelemeli bir fonksiyonun tam olarak nasıl çalıştığını anlamak, bunları ilk öğrendiğinizde biraz zor olabilir. Bir örnek üzerinde bir özyinelemeli fonksiyonun çalışmasını takip etmek, anlamanıza yardımcı olabilir. Önceki bölümdeki programa göz atalım. Diyelim ki kullanıcı klavyeden 4 sayısını girdi. Bu program çalışmaya başladığında, çalışma zamanı yığınında bir modül ve ana fonksiyon için birer etkinleştirme kaydı olacaktır.

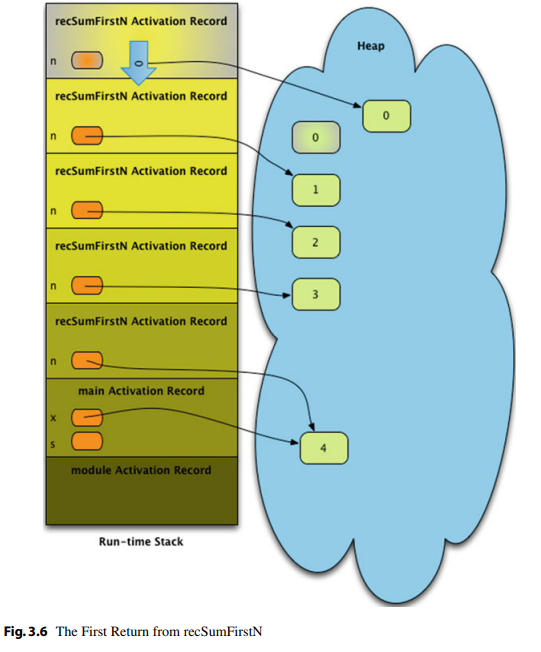
Program Sect. 3.4.2'deki kodun 10. satırına geldiğinde, recSumFirstN fonksiyonu ilk çağrıldığında, yeni bir etkinleştirme kaydı fonksiyon çağrısı için yığına itilecek ve bu durumda çalışma zamanı yığında üç etkinleştirme kaydı oluşacaktır. Ardından Python yorumlayıcısı, n'nin Fig. 3.5'teki resimde gösterildiği gibi 4 sayısına işaret ettiği satır olan 2. satıra atlar. Fonksiyonun yürütülmesi devam eder. n'nin değeri sıfır değildir, bu nedenle Python, başka bir recSumFirstN fonksiyon çağrısı içeren 5. satırı yürütür. Bu, Python yorumlayıcısını çalışma zamanı yığında başka bir etkinleştirme kaydı daha itmesine neden olur ve yorumlayıcı tekrar 2. satıra atlar. Bu sefer n'nin değeri 3'tür. Ancak yine de bu sıfır değildir, bu nedenle 5. satır yürütülür ve n için yeni bir değer olan 2 ile başka bir etkinleştirme kaydı itilir. Bu işlem, n'nin 1 ve 0 değerleri için iki kez daha tekrarlanır.

Bu programın yürütülmesinde dikkat edilmesi gereken önemli bir nokta, her özyinelemeli fonksiyon çağrısı için n değişkeninin bir kopyası bulunmasıdır. Bir etkinleştirme kaydı, fonksiyonun yerel kapsamındaki tüm değişkenlerin ve parametrelerin lokal değerlerini tutar.



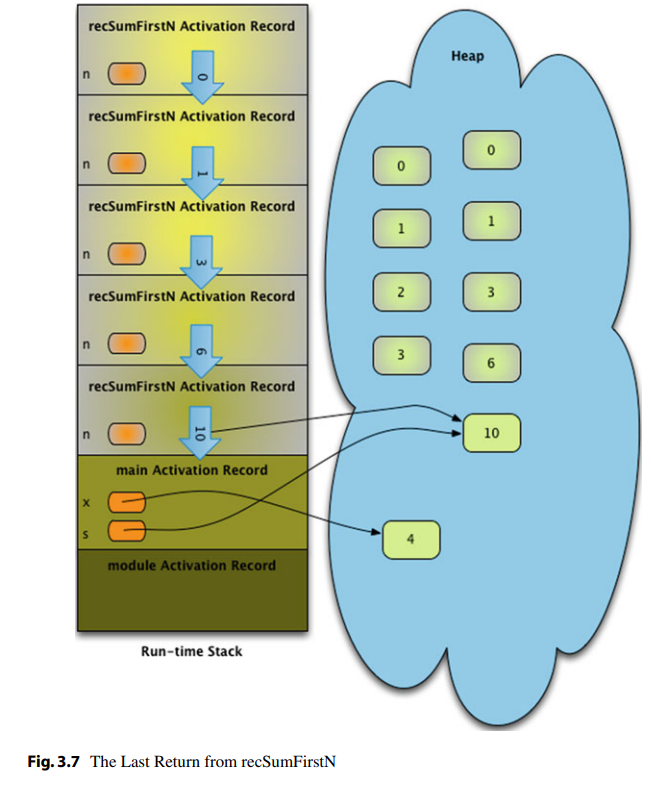
Her defasında fonksiyon çağrıldığında yeni bir etkinleştirme kaydı itilir ve yerel değişkenlerin yeni bir kopyası etkinleştirme kaydı içinde saklanır. Fig. 3.5'teki resim, çalışma zamanı yığınının en derin noktasını tasvir etmektedir.

Fonksiyonun yürütülmesi, n'nin 0 olduğu noktaya geldiğinde, Python yorumlayıcısı n'nin kodun 2. satırında 0 olduğunu bulur. İşte sumFirstN fonksiyonunun ilk değerini döndüğü nokta budur. n'nin değeri 1 olan önceki fonksiyon çağrısına 0'ı döndürür. Bu, kodun 5. satırında gerçekleşir. n'nin 0 olduğu fonksiyon çağrısı için etkinleştirme kaydı, çalışma zamanı yığınından çıkarılır. Bu, şekildeki etkinleştirme kaydının gölgelendirilmesiyle Fig. 3.6'da tasvir edilmiştir. Fonksiyon döndüğünde, etkinleştirme kaydının alanı daha sonra kullanılmak üzere geri alınır.



Heap üzerindeki 0 değerini içeren nesne, artık üzerine hiç referans olmadığından çöp toplayıcı tarafından da geri alınır. RecSumFirstN fonksiyonunun ilk dönüşünden sonra, Python yürütücüsü önceki fonksiyon çağrısının 5. satırına geri döner. Ancak, bu ifade de bir return ifadesi içerir.

Bu nedenle, fonksiyon tekrar döner. Bu sefer değeri 1 ile döner. Fonksiyon tekrar döner, ancak bu sefer değeri 3 ile döner. 6 değeri ile dönmesi için fonksiyon tekrar döner. Son olarak, fonksiyon tekrar döner, bu sefer 10 değeri ile döner. Ancak bu sefer recSumFirstN fonksiyonu, s'nin değerini 10'a işaret etmesi için ana fonksiyonun 10. satırına geri döner. Bu, Şekil 3.7'de tasvir edilmiştir.



Program, ekrana 10'u yazdıktan sonra ve 12. satırdan sonraki ana fonksiyondan, 15. satırdan sonraki modülden dönerek sona erer. Bu örnek, her recSumFirstN fonksiyonuna yapılan her özyinelemeli çağrının kendi n değişkeninin bir kopyasına sahip olduğunu, çünkü bu değişken recSumFirstN fonksiyonunun kapsamına özgü olduğunu göstermek açısından önemlidir. Fonksiyon her çağrıldığında, yerel değişkenler ve parametreler ilgili etkinleştirme kaydına kopyalanır. Bir fonksiyon çağrısı gerçekleştikten sonra ilgili etkinleştirme kaydı çalışma zamanı yığınının üzerinden çıkartılır. İşte özyinelemeli bir fonksiyonun nasıl yürütüldüğü budur.