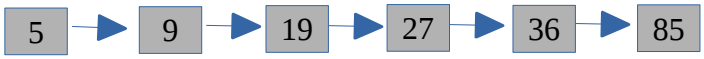
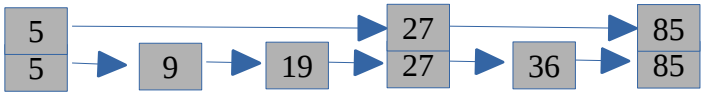
Skip-list (Atlamalı Liste)

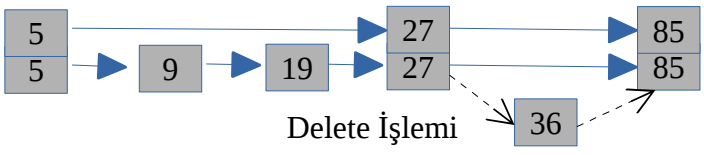
Linked List'e daha hızlı erişmek için kullanılan bir veri yapısıdır.
Bunu yapabilmek için listenin başında index oluşturur.
Aranan veriyi bulma, veri silme ve ekleme konularında diğer yapılardan daha başarılıdır.



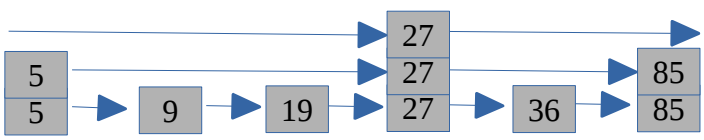
Mesela yukarıdaki sıralı sayılar üzerinden konuşacak olursak;
bu sayılar bir array yapısında tutulursa binary search bize oldukça hızlı bir erişim sağlar fakat bu sayılar array üzerinde bulunursa silme ve yeni sayı ekleme işlemleri tüm elemanları kaydırma zorunluluğundan dolayı zor olabilir.
Eğer listemiz üzerinde linear search ile arama yaparsak her elemana tek tek bakılacağı için istediğimizi bulmak zor olacaktır.
Bu sayıları binary tree yapısında tutarsak eleman bulma işlemi kolay olsa da silme konusunda yine sorunlar çıkacaktır.
Bunların yerine skip-list kullanırsak sorunlar çözülecektir.



Skip-listte'rastgele elemanlardan ikinci seviyede linked list oluşturulur. Bu listede elemanların bazıları bulunur. Yani liste atlamalı olarak oluşturulur.
Farklı bir seviye oluştuğu için bazı sayılar birden fazla pointera sahip olurlar.
Search:Bir sayı aranırken üst listede bir iterator sayılara teker teker bakar ve iteratorde 2 adet pointer bulunur. Bu sayede sayının aralığı belirlenebilir. Örneğin 36 değerini üst seviyede arayan bir iterator 3 sayıya bakarak 36 sayısının 27 ve 85 sayıları arasında olduğunu belirleyebilir. Daha sonra alt seviyeye inilerek sayı bulunur.
İnsert: Bir sayı eklenirken arama mantığı ile aynı şekilde ilk önce üst seviyelerden eklenecek yerin aralığı bulunur. Daha sonra bu aralıkta alt seviyelere inilerek eklenecek yer bulunur. İki sayının arasına eklenir ve sonraki değerin pointerını alır. Sonraki değerin pointerı da eklenen değerden onu gösterir. Örneğin, 8 sayısını ekleyeceğimiz zaman önce ikinci seviyeden 5 ile 27 arasında olduğunu buluruz. Sonra alt seviyeye inerek 9'a ulaştığımızda ekleyeceğimiz yeri tam olarak buluruz. 8 sayısı 5 ile 9'un arasına girer. 9'un adresini içeren pointer 8'i gösterir ve 9'un pointerı 8'den 9'a gelir.
Delete: Bir sayı silinirken sayı yine aynı şekilde bulunur. Sayı çıkarılır ve sayının adresini içeren pointer artık bu sayıdan sonra gelen sayının adresini içerir. Örneğin 36 sayısını silmek istersek önce sayının 27 ile 85 arasıda olduğunu buluruz ve alt seviyeye ineriz. Sonra bu sayıyı sileriz ve pointerını 85'e bağlarız.



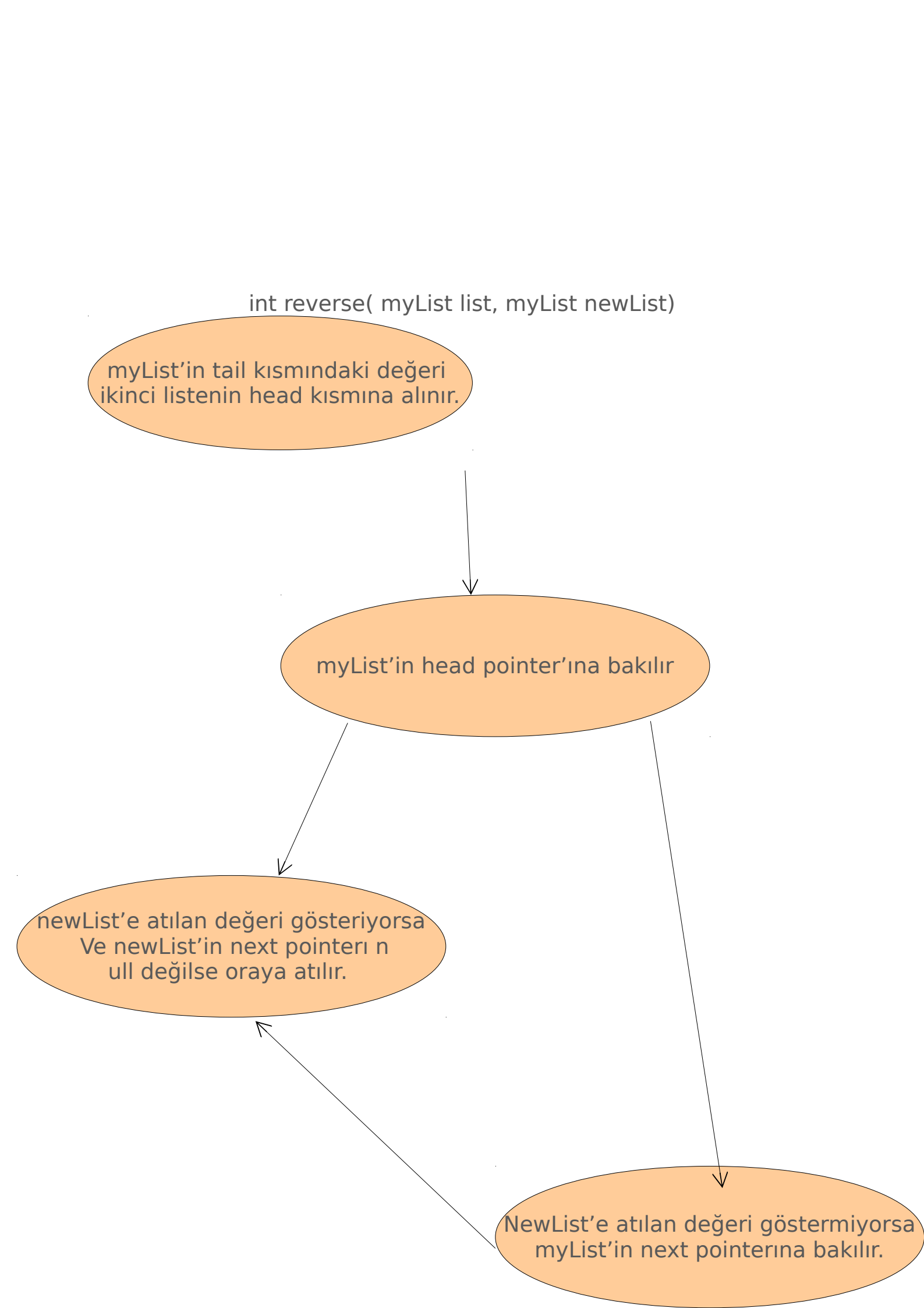
Delete İşlemi



Bu işlemler daha üst seviyelerde de gerçekleştirilebilir. Örneğin 19 sayısının üçüncü seviyede 27'den önce olduğu görülür ve bir alt seviyeye inilir. Daha sonra 5 ile 27 arasında olduğu görülür ve birinci seviyeye inilip burada bulunur.

int countInList(myList list,int key)

int firstRepeating(myList list)


int reverse(myList list, myList newList)
myList’in tail kısmındaki değeri
ikinci listenin head kısmına alınır.
myList’in head pointer’ına bakılır
newList’e atılan değeri gösteriyorsa
Ve newList’in next pointerı n
ull değilse oraya atılır.
NewList’e atılan değeri göstermiyorsa
myList’in next pointerına bakılır.