

Working of List in Python

Advantages

- Random access
- Cache Friendly

Disadvantages

- Insertion, Deletion are slow
- Search is also slow for unsorted

How does Dynamic size work?

- ⇒ Pre-allocate some extra space
- ⇒ If becomes full, do the following
 - Allocate a new space of larger size (Multiply by x)
 - Copy old space to new
 - Free old space

Analysed Time

- ⇒ Initial capacity: n
- ⇒ We double the size when become full
- ⇒ Average time to append (n + 1) items = $\Theta(1)$