Import xmlrpclib

XML-RPC

XML-RPC is a simple, portable way to make remote procedure calls over HTTP. It can be used with Perl, Java, Python, C, C++, PHP and many other programming languages.
Implementations are available for Unix, Windows and the Macintosh.

Note

The xmlrpclib module has been renamed to xmlrpc.client in Python 3
Xmlrpclib is the standard lib

Import xlrd

This package is for reading data and formatting information from older Excel files (ie: .xls).
Library for developers to extract data from Microsoft Excel (tm) spreadsheet files.
Extract data from Excel spreadsheets (.xls and .xlsx, versions 2.0 onwards) on any platform. Pure Python (2.6, 2.7, 3.2+). Strong support for Excel dates. Unicode-aware.

```
workbook = xlrd.open_workbook('sheet_name.xls')
worksheet = workbook.sheet_by_name(sheet1')
num_rows = worksheet.nrows - 1
num_cells = worksheet.ncols - 1
curr_row = 1
```

(The name in colored are the method name)

Import xlwt

This package is for writing data and formatting information to older Excel files (ie: .xls).

sock_common = xmlrpclib.ServerProxy('http://localhost:8069/xmlrpc/**common**')
uid = sock_common.login(dbname, username, pwd)
Common means (This is use for temporary connection)

sock = xmlrpclib.ServerProxy('http://localhost:8069/xmlrpc/**object**')
Object means (we can perform the operation method like search,read,write,create)

Example = sock.execute(dbname, uid, pwd, 'account.account', 'create', account_account)

browse() method is not supported by xmlrpc.

Starting the script

```
while curr_row < num_rows:
        curr_row += 1
        row = worksheet.row(curr_row)
```

Xlsx workbooks have no macro, unlike a xlsm file. Extension of the working area. In a Xls workbook, the row limit is **65,536** ($2^{16)}$ and **256 columns** ($2^{8)}$ which corresponds to the column IV. Now with xlsx workbooks (and xlsm), limits are **1,048,576** rows ($2^{20)}$ and **16,384** columns ($2^{14)}$ or the column XFD

http://www.studytonight.com/data-structures/stack-data-structure

Constructor

A *constructor* is a special kind of method that Python calls when it instantiates an object using the definitions found in your class
The name of a constructor is always the same, `__init__()`

When you create a class without a constructor, Python automatically creates a default

It's overriding. Python does not support overloading.

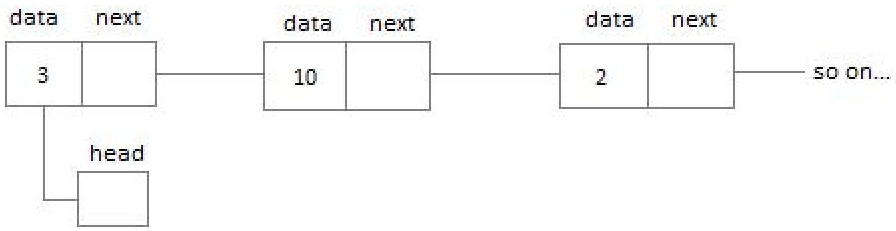
Queue Data Structures

first element is inserted from one end called REAR(also called tail), and the deletion of exisiting element takes place from the other end called asFRONT(also called head). This makes queue as FIFO data structure, which means that element inserted first will also be removed first.

The process to add an element into queue is called Enqueue and the process of removal of an element from queue is called Dequeue.

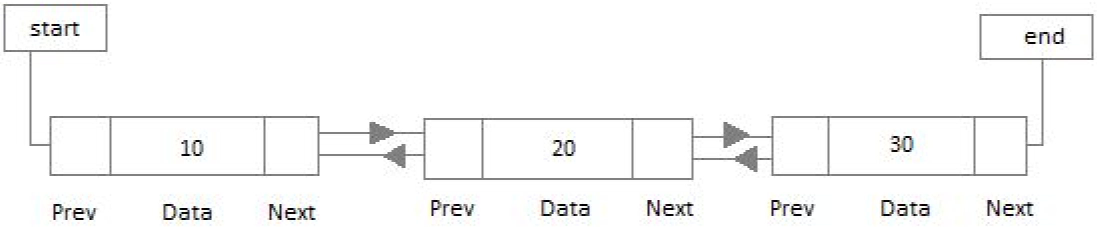
Types of Linked Lists

- **Singly Linked List : Singly linked lists contain nodes which have a data part as well as an address part i.e. next, which points to the next node in sequence of nodes. The operations we can perform on singly linked lists are insertion, deletion and traversal.**

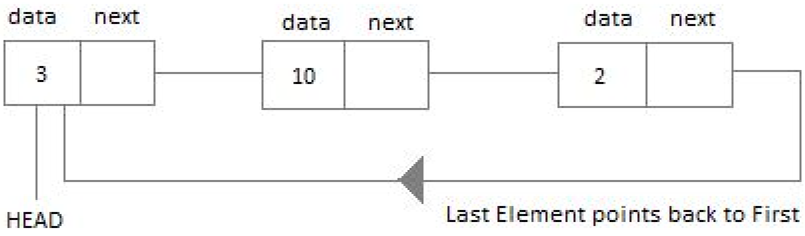


-

- **Doubly Linked List : In a doubly linked list, each node contains two links the first link points to the previous node and the next link points to the next node in the sequence.**



-
- **Circular Linked List : In the circular linked list the last node of the list contains the address of the first node and forms a circular chain.**



-
-

Bubble Sort compares all the element one by one and sort them based on their values.

Insertion Sorting

It is a simple Sorting algorithm which sorts the array by shifting elements one by one.

It is better than Selection Sort and Bubble Sort algorithm
Insertion Sort is adaptive, that means it reduces its total number of steps if given a partially sorted list, hence it increases its efficiency.

Selection Sorting

Selection sorting is conceptually the most simplest sorting algorithm. This algorithm first finds the smallest element in the array and exchanges it with the element in the first position, then find the second smallest element and exchange it with the element in the second position, and continues in this way until the entire array is sorted.

Quick Sort Algorithm

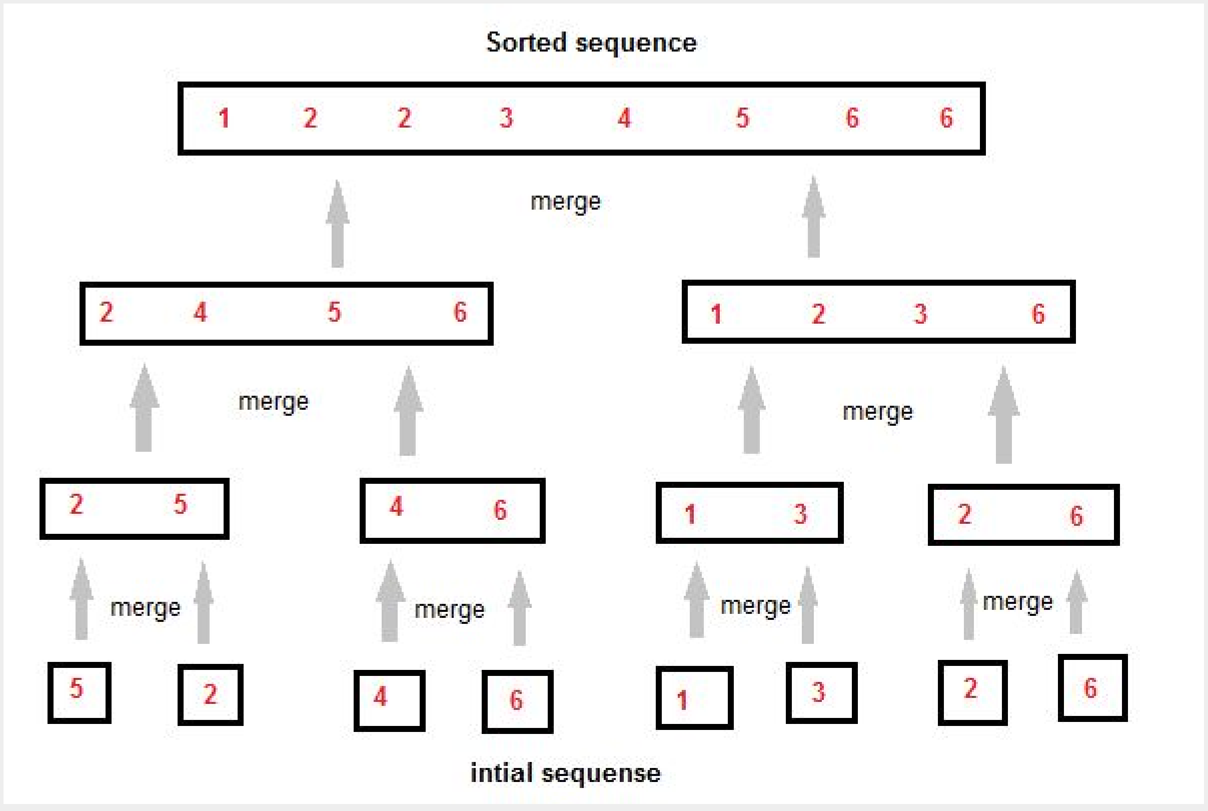
Quick Sort, as the name suggests, sorts any list very quickly. Quick sort is not stable search, but it is very fast and requires very less aditional space. It is based on the rule of **Divide and Conquer**(also called *partition-exchange sort*). This algorithm divides the list

`6 8 17 14` **`25`** `63 37 52`

Hnece after the first pass, pivot will be set at its position, with all the elements smaller to it on its left and all the elements larger than it on the right. Now `6 8 17 14` and `63 37 52` are considered as two separate lists, and same logic is applied on them, and we keep doing this until the complete list is sorted.

Merge Sort Algorithm

How Merge Sort Works



Like we can see in the above example, merge sort first breaks the unsorted list into sorted sublists, and then keep merging these sublists, to finlly get the complete sorted list.