

Algorithms and data structures

- How do you find efficient algorithms?
 - Hard to invent new ones
 - Easier to reduce problems to known solutions
 - Understand inherent complexity of problem
 - Think about how to break problem into sub-problems
 - Relate sub-problems to other problems for which there already exist efficient algorithms

Search algorithms

- Search algorithm – method for finding an item or group of items with specific properties within a collection of items.
- Collection called the search space
- Saw examples – finding square root as a search problem
 - Exhaustive enumeration
 - Bisection search
 - Newton-Raphson

Linear search and indirection

- Simple search method

```
def search(L, e):  
    for i in range(len(L)):  
        if L[i] == e:  
            return True  
    return False
```

- Complexity?
 - If element not in list, $O(\text{len}(L))$ tests
 - So at best linear in length of L

Linear search and indirection

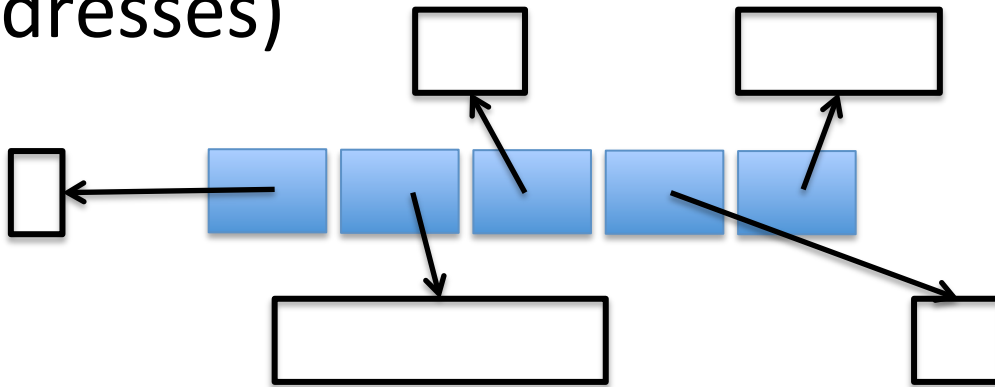
- Why “at best linear”?
 - Assumes each test in loop can be done in constant time
 - But does Python retrieve the i^{th} element of a list in constant time?

Indirection

- Simple case: list of ints
 - Each element is of same size (e.g., four units of memory – or four eight bit bytes)
 - Then address in memory of i^{th} element is $\text{start} + 4 * i$ where start is address of start of list
 - So can get to that point in memory in constant time

Indirection

- But what if list is of objects of arbitrary size?
- Use indirection
- Represent a list as a combination of a length (number of objects), and a sequence of fixed size pointers to objects (or memory addresses)



Indirection

- If length field is 4 units of memory, and each pointer occupies 4 units of memory
- Then **address** of i^{th} element is stored at $\text{start} + 4 + 4 * i$
- This address can be found in constant time, and value stored at address also found in constant time
- So search is linear
- **Indirection** – accessing something by first accessing something else that contains a reference to thing sought