Searching and Sorting using Lists and Arrays
Binary Search

- "Divide and Conquer" strategy
  - at every stage, we reduce the size of the problem to half the earlier stage
- Strategy: Compare with the middle element of current range, and eliminate half of the range

```python
# Algorithm Binary Search
def binarysearch(ar, l, r, x):
    while l<=r:
        mid = l+(r-l)//2
        if ar[mid] == x:
            return mid
        elif ar[mid] < x:
            l = mid + 1
        else:
            r = mid - 1
    return -1
```
# Algorithm Binary Search

def binarysearch(ar, l, r, x):
    if r >= l:
        mid = l + (r - l)//2
        if ar[mid] == x:
            return mid
        elif ar[mid] > x:
            return binarysearch(ar, l, mid-1, x)
        else:
            return binarysearch(ar, mid+1, r, x)
    else:
        return -1
Sorting an Array

• Rearranging array contents in increasing or decreasing order

How do we sort?

Sort in increasing order

How do we sort?

Courtesy Prof P R Panda CSE, IIT Delhi
Simple Sorting Algorithm

```python
for i in range(len(A)):
    k = position of min. element between A[i] and A[N-1]
    Swap A[i] and A[k]
```
Simple Sorting Algorithm

for in range(len(A)):
    k = position of min. element between A[i] and A[N-1]
    Swap A[i] and A[k]


for j in range(i+1, len(A)):
    if A[min_index] > A[j]:
        min_index = j

t = A[i]
A[i] = A[k]
A[k] = t