Searching

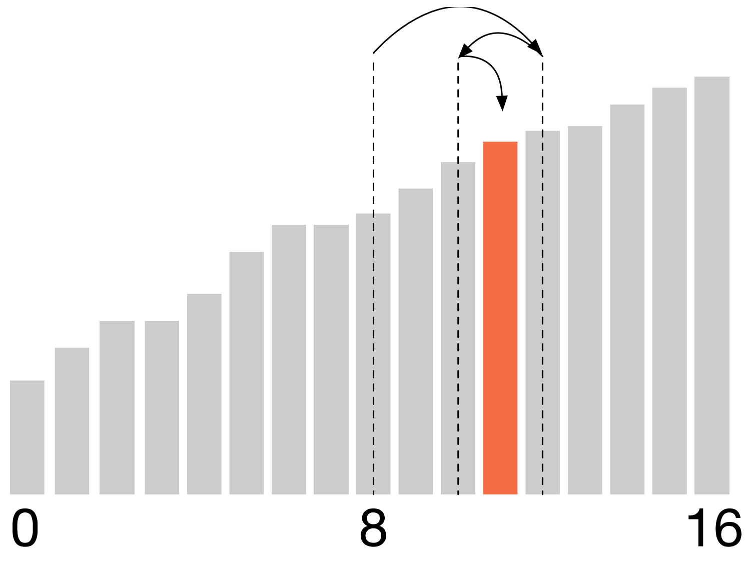
Mustafa Hajij
University of San Francisco

UNIVERSITY OF SAN FRANCISCO

Binary search (review sort of)

- If we know data is sorted, we can search much faster than linearly
- Means we don't have to examine every element even worst-case

```
def binsearch(a,x):
    left = 0; right = len(a)-1
    while left<=right:
        mid = (left + right)//2
        if a[mid]==x: return mid
        if x < a[mid]: right = mid-1
        else: left = mid+1
    return -1
```



See http://interactivepython.org/runestone/static/pythonds/SortSearch/TheBinarySearch.html

Compare to (tail-)recursive version

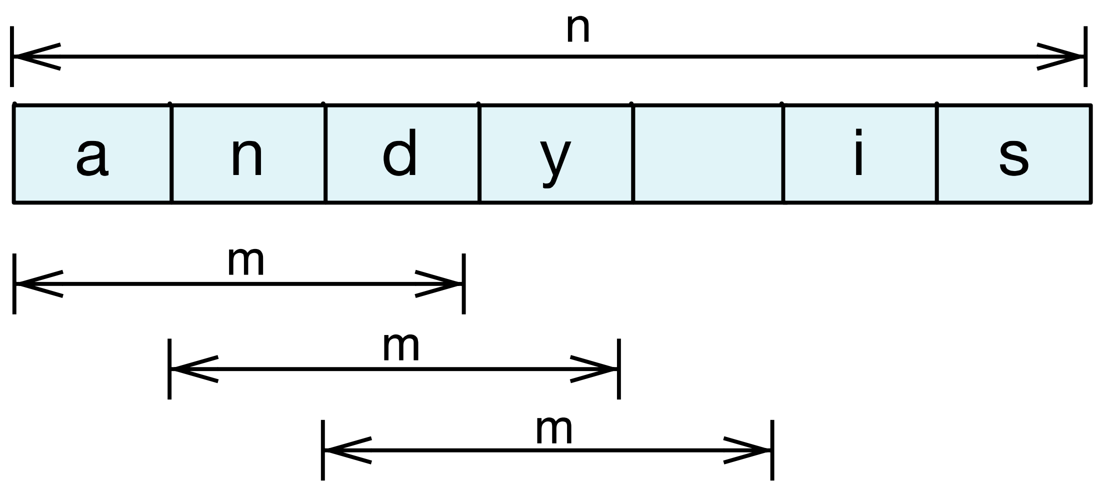
```
def binsearch(a,x,left,right):
    if left > right: return -1
    mid = (left + right)//2
    if a[mid]==x: return mid
    if x < a[mid]:
        return binsearch(a,x,left,mid-1)
    else:
        return binsearch(a,x,mid+1,right)
```

```
left = 0; right = len(a)-1
while left<=right:
    mid = (left + right)//2
    if a[mid]==x: return mid
    if x < a[mid]: right = mid-1
    else: left = mid+1
```

Bracket region with element

String matching

- **Problem**: Given a document of length n characters and a string of length m, find an occurrence or all occurrences
- Brute force algorithm is O(nm)
- Theoretical best-case algorithm exists for O(n + m)
- **Exercise**: Describe brute force algorithm; why is it ”slow”?



UNIVERSITY OF SAN FRANCISCO