

- Previous Lecture:

- Recursion

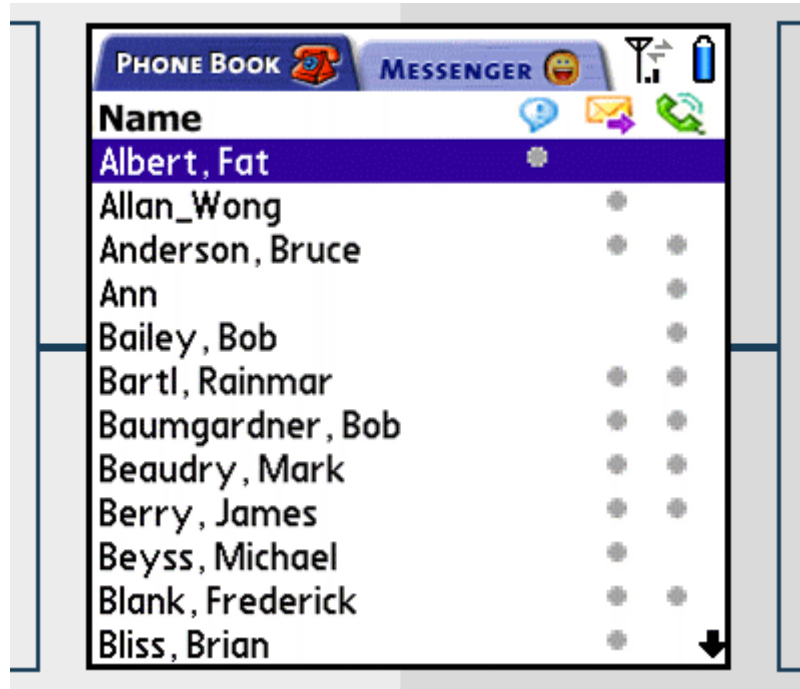
- Today's Lecture:

- Sorting and searching
 - *Insertion sort*, linear search
 - Read about *Bubble Sort* in Insight
- “Divide and conquer” strategies
 - Binary search

- Announcements

- Discussion in computer lab this week
- **P6** due Thursday at 11pm
- **Final exam:** Dec 7th 2pm for both Lec1 and Lec 2

Sorting data allows us to search more easily



Boston Marathon Top Women Finishers

				Official Time	State	Country	Ctz
				2:25:25		ETH	
				2:25:27		RUS	
				2:26:34		KEN	
				2:28:12		LAT	
				2:29:48		ETH	
				2:30:52		ITA	
				2:33:56		ROM	
				2:34:37		ETH	
				2:35:37		RUS	
				2:44:44	IL	USA	CAN
				2:45:54	NS	CAN	
				2:46:25		KEN	
				2:47:17	FL	USA	RUS
				2:47:36		AUS	
				2:48:43	MN	USA	

Name	Score	Grade
Jorge	92.1	
Ahn	91.5	
Oluban	90.6	
Chi	88.9	
Minale	88.1	
Ball	87.2	

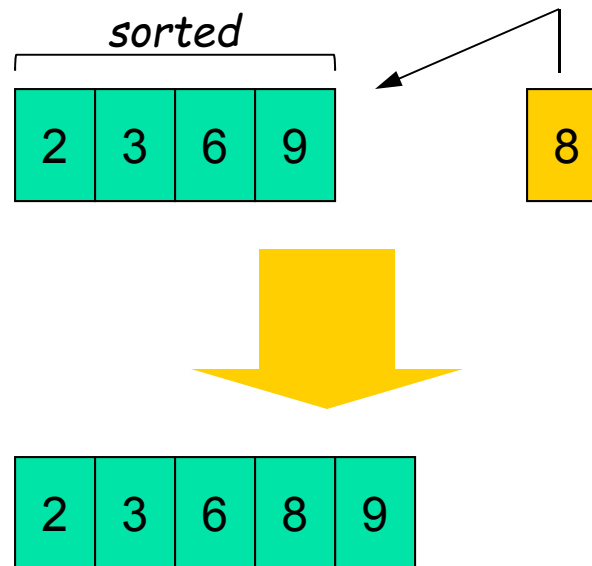
7	F12	Olaru, Nuta
8	F6	Guta, Robe Tola
9	F1	Grigoryeva, Lidiya
	F35	Hood, Stephanie A.
	F14	Robson, Denise C.
	F11	Chemjor, Magdaline
	F101	Sultanova-Zhdanova, Firaya
	F15	Mayger, Eliza M.
	F24	Anklam, Ashley A.

There are many algorithms for sorting

- Insertion Sort (to be discussed today)
 - Bubble Sort (read *Insight* §8.2)
 - Merge Sort (to be discussed Thursday)
 - Quick Sort (a variant used by Matlab's built-in `sort` function)
-
- Each has advantages and disadvantages. Some algorithms are faster (**time-efficient**) while others are **memory-efficient**
 - *Great opportunity for learning how to analyze programs and algorithms!*

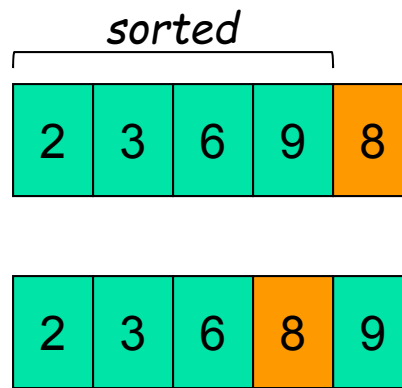
The Insertion Process

- Given a sorted array x , insert a number y such that the result is sorted



Insertion

one insert
process



Insert 8 into the sorted segment

Just swap 8 & 9

Insertion

one insert
process

2	3	6	9	8
---	---	---	---	---

2	3	6	8	9
---	---	---	---	---

sorted

2	3	6	8	9	4
---	---	---	---	---	---

Insert 4 into the sorted segment

Insertion

one insert
process

2	3	6	9	8
---	---	---	---	---

2	3	6	8	9
---	---	---	---	---

2	3	6	8	9	4
---	---	---	---	---	---

Compare adjacent components:
swap 9 & 4

Insertion

one insert
process

2	3	6	9	8
---	---	---	---	---

2	3	6	8	9
---	---	---	---	---

2	3	6	8	9	4
---	---	---	---	---	---

2	3	6	8	4	9
---	---	---	---	---	---

Compare adjacent components:
swap 8 & 4

Insertion

one insert
process

2	3	6	9	8
---	---	---	---	---

2	3	6	8	9
---	---	---	---	---

2	3	6	8	9	4
---	---	---	---	---	---

2	3	6	8	4	9
---	---	---	---	---	---

2	3	6	4	8	9
---	---	---	---	---	---

Compare adjacent components:
swap 6 & 4

Insertion

one insert
process

2	3	6	9	8
---	---	---	---	---

2	3	6	8	9
---	---	---	---	---

one insert
process

2	3	6	8	9	4
---	---	---	---	---	---

2	3	6	8	4	9
---	---	---	---	---	---

2	3	6	4	8	9
---	---	---	---	---	---

2	3	4	6	8	9
---	---	---	---	---	---

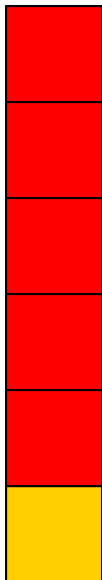
Compare adjacent components:
DONE! No more swaps.

See `Insert.m` for the insert process

Sort vector **x** using the **Insertion Sort** algorithm

Need to start with a *sorted* subvector. How do you find one?

x



Length 1 subvector is “sorted”

Insert **x**(2): [**x**(1:2),C,S] = Insert(**x**(1:2))

Insert **x**(3): [**x**(1:3),C,S] = Insert(**x**(1:3))

Insert **x**(4): [**x**(1:4),C,S] = Insert(**x**(1:4))

Insert **x**(5): [**x**(1:5),C,S] = Insert(**x**(1:5))

Insert **x**(6): [**x**(1:6),C,S] = Insert(**x**(1:6))

InsertionSort.m

Insertion Sort vs. Bubble Sort

- Read about Bubble Sort in *Insight* §8.2
- Both algorithms involve the repeated comparison of adjacent values and swaps
- Find out which algorithm is more efficient on average

Other efficiency considerations

- Worst case, best case, average case
 - Use of subfunction incurs an “overhead”
 - Memory use and access
-
- Example: Rather than directing the *insert* process to a subfunction, have it done “in-line.”
 - Also, Insertion sort can be done “in-place,” i.e., using “only” the memory space of the original vector.

```
function x = InsertionSortInplace(x)
% Sort vector x in ascending order with insertion sort

n = length(x);
for i= 1:n-1
    % Sort x(1:i+1) given that x(1:i) is sorted

end
```

```

function x = InsertionSortInplace(x)
% Sort vector x in ascending order with insertion sort

n = length(x);
for i= 1:n-1
    % Sort x(1:i+1) given that x(1:i) is sorted
    j= i;

    while

        % swap x(j+1) and x(j)

        j= j-1;

    end
end
end

```

Sort an array of objects

- Given `x`, a 1-d array of `Interval` references, sort `x` according to the widths of the `Intervals` from narrowest to widest
- Use the insertion sort algorithm
- How much of our code needs to be changed?

A. No change

B. One or two statements

C. About half the code

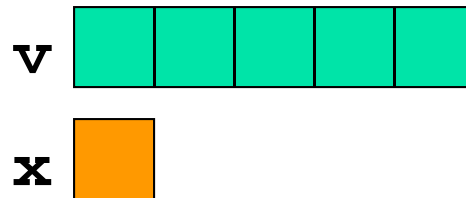
D. Most of the code

The only change is
in how we do the
comparison!

See `InsertionSortIntervals.m`

Searching for an item in an unorganized collection?

- May need to look through the whole collection to find the target item
- E.g., find value x in vector v



- Linear search

```
% Linear Search
```

```
% f is index of first occurrence
```

```
% of value x in vector v.
```

```
% f is -1 if x not found.
```

```
k= 1;
```

```
while k<=length(v) && v(k)~=x
```

```
    k= k + 1;
```

```
end
```

```
if k>length(v)
```

```
    f= -1; % signal for x not found
```

```
else
```

```
    f= k;
```

```
end
```

v	12	35	33	15	42	45
x	31					

```
% Linear Search
```

```
% f is index of first occurrence
```

```
% of value x in vector v.
```

```
% f is -1 if x not found.
```

```
k= 1;
```

```
while k<=length(v) && v(k)~=x
```

```
    k= k + 1;
```

```
end
```

```
if k>length(v)
```

```
    f= -1; % signal for x not found
```

```
else
```

```
    f= k;
```

```
end
```

A. squared

B. doubled

C. the same

D. halved

Suppose another vector is twice as long as v. The expected “effort” required to do a linear search is ...

```
% Linear Search
```

```
% f is index of first occurrence
```

```
% of value x in vector v.
```

```
% f is -1 if x not found.
```

```
k= 1;
```

```
while k<=length(v) && v(k)~=x
```

```
    k= k + 1;
```

```
end
```

```
if k>length(v)
```

```
    f= -1; % signal for x not found
```

```
else
```

```
    f= k;
```

```
end
```

Searching in
a sorted list should
require less work

v	12	15	33	35	42	45
x	31					

What if **v** is sorted?

An ordered (sorted) list

The Manhattan phone book has 1,000,000+ entries.

How is it possible to locate a name by examining just a tiny, tiny fraction of those entries?

wide at SuperPages.com

195

Car

C

17 566-1282	Cartage New England Inc 26 Allen Ln Ipswich 01938.....	978 356-9960	Carter F 24 Hillock Ros 02131.....	617 327-1105	Carter Nella E 333 Maschots Av Bos 02115.....	617 267-6483
81 447-4101	Cartagema Lydia 18 Jewett Ros 02131.....	617 323-7639	Faye & Ricky 357 Columbus Av Bos 02116.....	617 437-7331	Nicholas S F 115 Randolph Av Mil 02186.....	617 698-5307
800 257-9981	Cartagema Avith 9 Bancroft Ros 02119.....	617 442-9780	Francis S 134 Temple W Ros 02132..	617 323-6781	Nick 21 Fairfield Bos 02116.....	617 267-5222
17 566-1282	B Hyd 02136.....	617 361-5253	Franklin & Anne 221 Mt Auburn Cam 02138.....	617 354-0798	Nick & Debbi 196 Herrick Rd Newton 02459.....	617 527-0480
17 364-5188	Jessica 50 Decatur Cha 02129.....	617 241-0152	Fred 42 Haverford Jam 02130.....	617 524-3078	Nicole.....	617 698-0713
361-0380	Lucilla 174 Harvard Cam 02139.....	617 491-5621	Fred 96 Hincley Rd Mil 02186.....	617 698-1343	Norman G 38 Chickatawbut Dor 02122.....	617 822-1203
17 566-4548	M 95 Rowe Ros 02131.....	617 323-9713	G & R 8 Verdun Dor 02124.....	617 436-8906	P 94 Crestwood Pk Ros 02121.....	617 427-4754
17 628-8248	Melvin 501 Green Cam 02139.....	617 576-1061	G T 27 Franklin Av Som 02145.....	617 623-7121	P E 501 E Sixth S Bos 02127.....	617 268-4213
17 445-5116	Carte Nicholas 18 Appleton Boston 02116.....	617 695-6996	Gayle 25 Frontenac Dor 02124.....	617 825-0322	P L 44 Hutchings Ros 02121.....	617 427-9170
17 822-2982	Cartegena O 4 Millard Bos 02118.....	617 338-8219	George 125 Nashua Bos 02114.....	617 367-9548	P R 91 Byrner Jam 02130.....	617 983-8692
17 427-5712	Carten Thos J Sr & Claire 1 Paradise Rd Mil 02186.....	617 698-6163	Cartar Halliday Associate 107 S Street Bos 02111.....	617 456-1689	Paul & Constance 114 Anawan Av W Ros 02132.....	617 325-2036
17 569-2698	Thomas & Kathleen 50 Thompson Ln Mil 02186.....	617 696-6919	Cartar Harry F 26 Rung Bk Rd W Ros 02132.....	617 325-5465	Paul E 501 E Sixth St S Bos 02127.....	617 268-4546
17 667-5190	Cartar A Ros 02131.....	617 327-2257	Cartar Hide Co Inc 146 Summer Bos 02110.....	617 542-7987	Paul M 27 Union Bri 02135.....	617 787-2115
17 569-1417	A Rosbury.....	617 442-5230	Cartar Hilary 61 Harvey Cam 02140.....	617 876-2750	Cartar Pile Driving Inc 17 Beaver Ct Framingham 01702.....	Wellesley Telc 781 235-8488
17 338-9110	A 31 Bethune Wy Roxbury 02119.....	617 442-1219	Horace 241 Walnut Av Rosbury 02119.....	617 442-5307	Cartar Prudence 46 Franklin Watertown 02172.....	617 393-3782
17 825-9195	A 260 Putnam Av Cambridge 02139.....	617 492-4174	Howard Jr 26 Notre Dme Ros 02119.....	617 445-5552	Prudence 46 Franklin Watertown 02172.....	617 926-7063
17 296-1593	Adams 361 Centre St Mil 02186.....	617 698-9074	J Cam.....	617 354-2688	Reginald 106 Brunswick Dorchester 02121.....	617 541-2843
17 670-2078	Alice 108 Kilmarock Bos 02215.....	617 425-0193	J 15 Chatham Bro 02446.....	617 232-7990	Renee & Andrew 10 Walnut Bos 02108.....	617 720-3765
17 623-9001	Alice 45 Market Cambridge 02139.....	617 945-2711	J 518 Harvard Bro 02446.....	617 730-9483	Cartar Rice Dowd Bulley Duntton Publishing 163 Main Wilmington 01887.....	800 638-1671
17 296-4725	Andrew F 62 Vinal Av Som 02143.....	617 625-7623	J 775 Ylv Pkwy West Roxbury 02132.....	617 323-5574	Toll Free-Dial '1' & Then.....	800 619-7447
17 542-1521	Cartar Anne MD 1101 Beacon Bro 02446.....	617 739-1022	Cartar J Jacques MD 1 Brookline Pl Bro 02446.....	617 735-8787	Cust Svc-Printing 613 Main Wilmington.....	800 648-7447
17 364-5232	Cartar Athens 272 Newbury Boston 02116.....	617 536-6329	Cartar J M 1410 Columbia Rd S Bos 02127.....	617 464-1040	Toll Free-Dial '1' & Then.....	978 988-7447
17 541-5649	B E 48 Gladeside Av Mat 02126.....	617 296-6911	Cartar J M Ornamental Ironworks Call.....	617 436-5353	Ingalls Cronin 163 Main Wilmington 01887.....	800 638-1673
17 739-2662	Cartar Barbara L MD Tufts-New England Medical Center Bos 02111.....	617 636-0051	Cartar J Veal Co 48 Newmarket Sq Ros 02118.....	617 442-1775	Toll Free-Dial '1' & Then.....	800 638-1673
17 879-0030	Cartar Becky Bos 02114.....	617 523-4368	Cartar James 1573 Cambridge St Cam 02138.....	617 492-1214	Cartar Richard 1079 Commwith Av Brighton 02215.....	617 987-0836
17 541-3948	Bernard J 112 Gladstone E Bos 02128.....	617 567-3430	James 182 Fisher Av Rosbury 02120.....	617 739-2193	Richard A 97 Mt Vernon Bos 02108.....	617 566-7293
17 436-1513	Bithiah 25 Medway Dor 02124.....	617 298-8713	James 37 Gold Star Rd Cambridge 02140.....	617 876-8841	Cartar Richard A MD 170 Commwith Av Bos 02116.....	617 267-0710
17 569-4119	Blake 26 Mt Vernon Bos 02108.....	617 367-9931	Jas L 14 Roseberry Rd Mat 02126.....	617 361-0773	Richard K 15 Mercer S Bos 02127.....	617 268-0448
800 569-8782	Cartar Broadcasting Co 20 Park Plz Bos 02116.....	617 423-0210	Jane 114 Adena Rd Newton 02465.....	617 964-0435	Robert L 175 Richdale Av Cam 02140.....	617 864-1535
	Cartar & Burgess Consultants Inc 23 East St Cam 02141.....	617 225-0200	Jeffrey 41 Warren Av Bos 02116.....	617 426-5994	Roger 150 St Botolph Bos 02115.....	617 424-6148
	C 228 Faywood Av East Boston 02128.....	617 569-1545	John 11 Mansfield Bri 02134.....	617 987-2163	Roy 44 Concord Av Cam 02138.....	617 491-6115
	C 359 Harvard Cam 02138.....	617 491-4822	John 327 Summer Bos 02110.....	617 423-4334	Royce 18 Seminary Cha 02129.....	617 241-0418
	C 610 Walk Hill Mat 02126.....	617 296-6392	John 40 Westwind Rd Dor 02125.....	617 282-1235		
	C & M 43 Burroughs Jam 02130.....	617 524-9558	June O 329 A Summit Av Bri 02135.....	617 734-6109		
			K 38 Browning Av Dorchester 02124.....	617 265-8456		
			K 17 Esmond Dorchester 02121.....	617 282-1593		

Key idea of “phone book search”: repeated halving

To find the page containing Pat Reed’s number...

```
while (Phone book is longer than 1 page)
    Open to the middle page.
    if “Reed” comes before the first entry,
        Rip and throw away the 2nd half.
    else
        Rip and throw away the 1st half.
    end
end
```

What happens to the phone book length?

Original:	3000	pages
After 1 rip:	1500	pages
After 2 rips:	750	pages
After 3 rips:	375	pages
After 4 rips:	188	pages
After 5 rips:	94	pages
:		
After 12 rips:	1	page

Binary Search

Repeatedly halving the size of the “search space” is the main idea behind the method of **binary search**.

An item in a sorted array of length **n** can be located with just **$\log_2 n$** comparisons.


```
% Linear Search
% f is index of first occurrence of value x in vector v.
% f is -1 if x not found.
k= 1;
while k<=length(v) && v(k)~=x
    k= k + 1;
end
if k>length(v)
    f= -1; % signal for x not found
else
    f= k;
end
```

n comparisons against the target
are needed in worst case,
n=length(v) .

Binary Search




Repeatedly halving the size of the “search space” is the main idea behind the method of **binary search**.

An item in a sorted array of length n can be located with just $\log_2 n$ comparisons.

“Savings” is significant!

n	$\log_2(n)$
100	7
1000	10
10000	13

Binary search: target $x = 70$

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98
												

L:

1

Mid:

6

R:




12

$v(\text{Mid}) \leq x$

**So throw away the
left half...**

Binary search: target $x = 70$

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98

L: 6

Mid: 9


R: 12

$x < v(\text{Mid})$

**So throw away the
right half...**

Binary search: target $x = 70$

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



L:

6

Mid:

7

R:

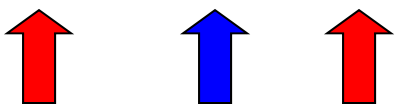
9

$v(\text{Mid}) \leq x$

**So throw away the
left half...**

Binary search: target $x = 70$

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



L: 7

Mid: 8

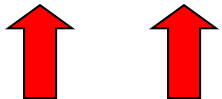
R: 9

$v(\text{Mid}) \leq x$

**So throw away the
left half...**

Binary search: target $x = 70$

	1	2	3	4	5	6	7	8	9	10	11	12
v	12	15	33	35	42	45	51	62	73	75	86	98



L:

8

Mid:

8

R:

9

Done because

$$R - L = 1$$

```
function L = binarySearch(x, v)
% Find position after which to insert x. v(1)<...<v(end).
% L is the index such that v(L) <= x < v(L+1);
% L=0 if x<v(1). If x>v(end), L=length(v) but x~=v(L).

% Maintain a search window [L..R] such that v(L)<=x<v(R).
% Since x may not be in v, initially set ...
L=0; R=length(v)+1;
```

```
% Keep halving [L..R] until R-L is 1,
% always keeping v(L) <= x < v(R)
while R ~= L+1
    m= floor((L+R)/2); % middle of search window
    if
        v(m) <= x
    else
        v(m) > x
    end
    if v(m) <= x
        L=m;
    else
        R=m;
    end
end
```



```

function L = binarySearch(x, v)
% Find position after which to insert x. v(1)<...<v(end).
% L is the index such that v(L) <= x < v(L+1);
% L=0 if x<v(1). If x>v(end), L=length(v) but x~=v(L).

% Maintain a search window [L..R] such that v(L)<=x<v(R).
% Since x may not be in v, initially set ...
L=0; R=length(v)+1;

% Keep halving [L..R] until R-L is 1,
% always keeping v(L) <= x < v(R)
while R ~= L+1
    m= floor((L+R)/2); % middle of search window
    if v(m) <= x

        L= m;
    else

        R= m;
    end
end
end

```

This version is different
from that in *Insight*

```

function L = binarySearch(x, v)
% Find position after which to insert x. v(1)<...<v(end).
% L is the index such that v(L) <= x < v(L+1);
% L=0 if x<v(1). If x>v(end), L=length(v) but x~=v(L).

% Maintain a search window [L..R] such that v(L)<=x<v(R).
% Since x may not be in v, initially set ...
L=0; R=length(v)+1;

% Keep halving [L..R] until R-L is 1,
% always keeping v(L) <= x < v(R)
while R ~= L+1
    m= floor((L+R)/2); % middle of search window
    if v(m) <= x
        L= m;
    else
        R= m;
    end
end
end

```

	20	30	40	46	50	52	68	70	
0	1	2	3	4	5	6	7	8	9

Play with `showBinarySearch.m`