
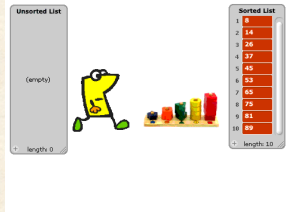


UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202 Introduction to Computation Professor Andrea Arpaci-Dusseau
Fall 2010


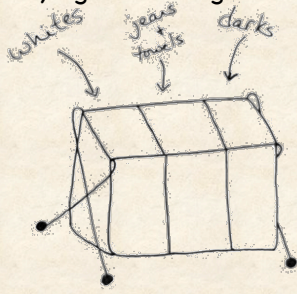
Lecture 36: How can computation... sort data in order for you?

"Now remember while we're in 'sort order', which one of you is supposed to choose ASCENDING and DESCENDING?"

What is sorting?


Sorting in common language has different meaning
This is not what we mean by sorting:
Not organizing or classifying into categories

What is sorting?

Put keys (and associated data) in specified order

- Ascending or descending
- Numerical or alphabetical



Sort keys and keep data with it

- Key: High score
 - Data: name
- Key: Employee id
 - Data: Name, Position, Phone, Salary
- Key: Search terms for web page
 - Data: URL, cached version, similar pages

Why is sorting important?

General:

- Can find what you want faster given sorted data

Sorted data is easier to search through

- Can apply binary search instead of exhaustive search

Trivial to find minimum and maximum elements

- First and last in list

Easy to find duplicate values

- Adjacent to each other in list

Easy to find patterns, anomalies, gap

- TCP can find missing packets

Many Different Sorting Algorithms

Today: Slow algorithms, but easy to understand

- Selection sort
- Insertion sort

Next Lecture: Faster algorithms

- Mergesort
- Quicksort

Review: How to find Min in List?

```

when I receive Find Min
  set index to 1
  set min to item index of Unsorted List
  set index of min to index
  repeat length of Unsorted List
    if item index of Unsorted List < min
      set min to item index of Unsorted List
      set index of min to index
  change index by 1
  
```

Loop through List using index variable

Input:

- List : Unsorted List

Output:

- Min
- Min index

Local variable:

- index

Robust to length of List

How can you sort using Minimum?

Unsorted list

8
5
2
6
9
3
1
4
0
7

How can you sort list of numbers if you can find the minimum?

- Move numbers into "sorted list"

Sorted list

Algorithm 1: Selection Sort

Unsorted list

8
5
2
6
9
3
1
4
0
7

To sort data...

Repeat until nothing in unsorted list:

- Find minimum element
- Add element to sorted list
- Delete from unsorted list

Sorted list

Selection Sort in Action

Unsorted list	82	54	28	64	91	37	19	46	9	73
Sorted list										

How to Implement Selection Sort in Scratch?

Control code

```

when clicked
ask How large is the list? and wait
set List Length to answer
broadcast Make List and wait
say Please sort a list for me! for 2 secs
say join The list has join List Length elements for 2 secs
broadcast Sort List and wait
say Can you please check it too? for 2 secs
broadcast Check List and wait
if List Sorted? = 1
say Thank! It is sorted! for 2 secs
else
say Oh no! It is not sorted! for 2 secs
                    
```

Asks Sorter Sprite to do work

- Create unsorted list
- Get the list sorted
- Check that the list really is sorted

Sorter Sprite: Helper Functions

Make List

```

when I receive Make List
delete all of Unsorted List
repeat List Length
insert pick random 1 to List Length + 10 at 1 of Unsorted List
                    
```

Check List

```

when I receive Check List
set List Sorted? to 1
set i to 1
set previous to item i of Sorted List
repeat List Length
if item i of Sorted List < previous
set List Sorted? to 0
set previous to item i of Sorted List
change i by 1
                    
```

Selection Sorter Sprite

```

when I receive Sort List
delete all of Sorted List
repeat List Length
broadcast Find Min and wait
add min to Sorted List
delete index of min of Unsorted List
                    
```

```

when I receive Find Min
set index to 1
set min to item index of Unsorted List
set index of min to index
repeat length of Unsorted List
if item index of Unsorted List < min
set min to item index of Unsorted List
set index of min to index
change index by 1
                    
```

Selection Sort: Two Lists
 Finds minimum remaining element in unsorted
 Adds to Sorted (in order)
 Deletes minimum from unsorted

Selection Sort: Do we need two Lists?

Unsorted list	8	8	8	8	8	8	8	8	9	
	5	5	5	5	5	5	6	9	9	
	2	2	2	6	6	6	9	7		
	6	6	6	9	9	9	7			
	9	9	9	3	4	7				
	3	3	3	4	7					
	1	1	4	7						
	4	4	7							
	0	7								
	7									
Sorted list	0	0	0	0	0	0	0	0	0	
		1								
			2							
				3						
					4					
						5				
							6			
								7		
									8	
										9

Selection Sort: One List Demo

Sorted

Minimum

Checking

0
1
2
3
9
6
5
4
8
7

Swaps minimum with key at desired location

Selection Sort in Scratch: One List

Variable i:
Number of sorted elements

Variable j:
Looks for min of remaining unsorted elements (start at variable "i" in each iteration)

Algorithm 2: Insertion Sort

What algorithm do you use to sort cards?

Insertion Sort

Divide cards into two groups: sorted and unsorted

Initial state: 1 sorted card, N-1 unsorted

Repeat for all cards

- Remove 1st card from unsorted portion
- Insert into correct location in sorted list
 - Repeat loop
 - Keep moving down list until card to left < new card
 - Could change algorithm to start at bottom and move up...
- Change definition of sorted vs. unsorted portions

Insertion Sort in Scratch

```

when I receive Sort List
  set i to 1
  repeat until i = List Length
    change i by 1
    set j to i
    repeat until item j - 1 of Unsorted List < item j of Unsorted List or j = 1
      set tmp to item j - 1 of Unsorted List
      replace item j - 1 of Unsorted List with item j of Unsorted List
      replace item j of Unsorted List with tmp
    change j by 1
  
```

Repeat for all cards

- Take 1st unsorted card
- Insert into correct location in sorted list: Repeat loop
- Keep moving down list until card to left is smaller than new card OR at beginning of list

Which Sorting Algorithm is Best?

Compare number of loop iterations as function of N - size of input list

Previously analyzed searching algorithms

Linear search: $O(N)$ operations

Binary search: $O(\log_2 N)$ operations

Selection Sort: How many loop iterations?

Selection Sort: Two Lists

```

when I receive Sort List
  delete all of Sorted List
  repeat List Length
    set i to 1
    set min to item i of Unsorted List
    set index of min to i
    repeat length of Unsorted List
      if item i of Unsorted List < min
        set min to item i of Unsorted List
        set index of min to i
    change i by 1
  delete index of min of Unsorted List
  add min to Sorted List
  
```

Size of list: N

2 loops: Outer and inner

How many iterations of outer loop?

- N

How many iterations of inner loop?

- N, N-1, N-2, ... 1
- Average: N/2

Total?

- $N * N/2$

Complexity?

- $O(N^2)$

Insertion Sort in Scratch

```

when I receive Sort List
  set i to 1
  repeat until i = List Length
    change i by 1
    set j to i
    repeat until (item j - 1 of Unsorted List < item j of Unsorted List or j = 1)
      set tmp to item j - 1 of Unsorted List
      replace item j - 1 of Unsorted List with item j of Unsorted List
      replace item j of Unsorted List with tmp
      change j by -1
  
```

Outer loop?

Always N

Inner loop - Worst case? Data in reverse order! Must move key to beginning

1, 2, ..., N-3, N-2, N- → N/2

Best case? Data sorted already! Done immediately!

0

Average case? Move to middle of list... ½ * Worst case
 $O(N^2)$

Today's Summary

Intuitive but Slow Sorting

- Selection sort: **Select** minimum and make next in list
- Insertion sort: Take next and **insert** in correct place
- Both require operations $O(N^2)$
- Tip: Always write check code (easier than work code)

Reading

- Section 3.3.3

Announcements

- Exam 2 returned (Median 86, Ave 80)
- Project 2: Trivia Lists
- Click "Love It" for Project 1 samples