

Week 10

ASSIGNMENTS

h7 due before 10pm on Monday 4/1

h8 available soon and due before 10pm on 4/8

p4 available _____ and due before 10pm on Thursday 4/16

Peer Mentors: will help student teams with graph terminology, implementations, and algorithms, bring your own problems, or let our peer mentors suggest some

Module: Week 10 (start on week 11 before next week)

Read Team Project and install JavaFX before lecture Week 11

THIS WEEK

- Final Team Project (create your 4 person teams now)
- Sets
 - Notation
 - SetADT
 - Implementation
- Linear Sorts
 - Radix sort
 - Flash sort
- Graphic User Interfaces

NEXT WEEK

- Java FX
- more Java
-

Notation

https://www.rapidtables.com/math/symbols/Set_Symbols.html

 \emptyset
 \mathbb{N}
 \mathbb{R}
 $A \in \text{letters}$
 $|A|$
 $A \subseteq B$
 $A \not\subseteq B$
 $A \supseteq B$
 $A \not\supseteq B$
 A'
 $A \subset B$
 $A \not\subset B$
 $A \supset B$
 $A \not\supset B$

Basic Operations

 $A \cup B$
 $A \cap B$
 $A - B$
 $A \oplus B$
 $A \Delta B = (A \oplus B) \cup (B \oplus A)$

SetADT

Operations

`boolean add(E e)` - add if item is not present
`boolean contains(Object o)` - true iff o is present
`boolean remove(Object o)` - remove o if present
`boolean isEmpty()` - true if no element
`int size()` - returns number of elements

Complexity Analysis of Implementation Options

Assume: N is number of nodes and H is height of tree

	insert	lookup	remove	iteration
Array				
Sorted Array				
Linked List				
BST				
Balanced Search Tree				
Hash Table "good"				

De Morgan's Laws

If A and B are any two sets then:

$$(A \cup B)' = A' \cap B'$$

$$(A \cap B)' = A' \cup B'$$

Radix Sort

*

Pre-conditions:

number of items (N):

range of unique digits (RANGE):

length of item's sequence of digits (LEN)

Idea

Sort the following integers:

121 367 354 873 777 333 123 222 411 262 897

What is N? _____ RANGE? _____ LEN? _____

Pass 1:

0 1 2 3 4 5 6 7 8 9

Pass 2:

0 1 2 3 4 5 6 7 8 9

Pass 3:

0 1 2 3 4 5 6 7 8 9

Sorted:

Radix Sort: algorithm

We know:*Algorithm*

```
List[] digitQ = new List[RANGE]

for (i = 0; i < RANGE; i++):
    digitQ[i] = new Queue()

for (pos=LEN-1; pos >= 0; pos--):
    for (j=0; j < N; j++)
        let x = digit in position pos of the item in A[j]
        digitQ[x].enqueue(A[j])

    index = 0
    for ( j=0; j < RANGE; j++ ):

        while (!digitQ[j].isEmpty()):
            A[index] = digitQ[j].dequeue()
            index++
```

Complexity

Flashsort

Data and Variables

- A - the array of elements to be sorted
- n - the number of items to sort, A.length
- Amax - the maximum value in A
- Amin - the minimum value in A
- m - the number of bins to divide items into
- A[i] - the element at index i of A
- K(A[i]) - classification (bin@) for A_i
- L - an array use to track number of items in that bin
- L - later used to indicate next available index for that bin

Setup

Classification

Permutation

Insertion

Flashsort Example

<i>index</i>	0	1	2	3	4	5	6	7
K(A[i])	2	1	3	1	1	2	1	2
A =	5	3	7	1	2	6	1	4

INITIAL SETUP

K: [2,1,3,1,1,2,1,2]

A: [5,3,7,1,2,6,1,4]

N = 8

m = 3

Amin = 1

Amax = 7

indexMax = 2

$$K(A[i]) = 1 + (\text{int}) ((m-1) (A_i - \text{Amin}) / (\text{Amax} - \text{Amin}))$$

L (AFTER CLASSIFIED)

--	--	--	--

L (AFTER BOUNDARY INDICES SET)

--	--	--	--

// PERMUTATION ALGORITHM

swap first value in A with max value in A

set j to index 0

while j is less than last index of A:

set F to A[i]

get index I for F

if j is less than i:

Do:

Save the value from A[i] as temp (to be the next F)

Place F at A[i]

Decrement the boundary index in I = L[K(F)]

Assign temp to F to be the next value place

Get next I for F

Repeat while I is not j

Increment j

Graphic User Interfaces (GUIs)

Idea:

Java FX

AWT - abstract windowing toolkit (classes that create graphic UI components)

Applets - run java code in a web browser

Swing - encapsulated common gui components - easier to use than awt packages

Java FX - latest version, has support for desktop computers and web browsers

Why Java FX?

Install e(fx)clipse

1. Install JDK 8
2. Install Eclipse Oxygen or Proton
3. Install Java FX project wizard
 - a. launch Eclipse
 - b. Go to Help -> Install New Software
 - c. Search for "e(fx)clipse"
 - d. Install "e(fx)clipse" and wait patiently for installation to complete (a few minutes)
 - e. Accept certificate and User Agreement
 - f. Restart Eclipse as directed
4. Create a Java FX Project
 - a. Select File -> New -> Other
 - b. Select Java FX -> Java FX Project
 - c. Enter project details as before and proceed
5. Run your Java FX application
 - a. Open class Main
 - b. Click Run as Application
6. View the GUI that is built by default
7. Close View

Java FX Basics

Java FX Demo (in-class)