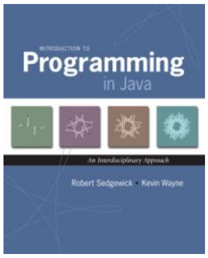
4.2 Sorting and Searching



Sorting


pentrust.org

Sorting


pentrust.org


shanghaiscrap.org

Sorting


pentrust.org


shanghaiscrap.org


De Beers

Sorting

Sorting problem. Rearrange N items in ascending order.

Applications. Statistics, databases, data compression, bioinformatics, computer graphics, scientific computing, (too numerous to list), ...

Hauser	→	Hanley
Hong		Haskell
Hsu		Hauser
Hayes		Hayes
Haskell		Hong
Hanley		Hornet
Hornet		Hsu

Insertion Sort


The Bridge Table

Insertion Sort

Insertion sort.
- Brute-force sorting solution.
- Move left-to-right through array.
- Exchange next element with larger elements to its left, one-by-one.

i	j	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
6	6	and	had	him	his	was	you	the	but
6	5	and	had	him	his	was	the	you	but
6	4	and	had	him	his	the	was	you	but
		and	had	him	his	the	was	you	but

Inserting a[6] *into position by exchanging with larger entries to its left*

Insertion Sort

Insertion sort.
- Brute-force sorting solution.
- Move left-to-right through array.
- Exchange next element with larger elements to its left, one-by-one.

i	j	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
		was	had	him	and	you	his	the	but
1	0	had	was	him	and	you	his	the	but
2	1	had	him	was	and	you	his	the	but
3	0	and	had	him	was	you	his	the	but
4	4	and	had	him	was	you	his	the	but
5	3	and	had	him	his	was	you	the	but
6	4	and	had	him	his	the	was	you	but
7	1	and	but	had	him	his	the	was	you
		and	but	had	him	his	the	was	you

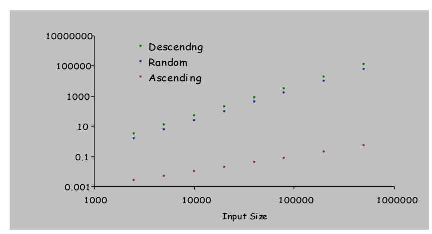
Inserting a[1] *through* a[N-1] *into position (insertion sort)*

Insertion Sort: Java Implementation

```java
public class Insertion {

   public static void sort(int[] a) {
      int N = a.length;
      for (int i = 1; i < N; i++)
         for (int j = i; j > 0; j--)
            if (a[j-1] < a[j])
               exch(a, j-1, j);
            else break;
   }

   private static void exch(int[] a, int i, int j) {
      int swap = a[i];
      a[i] = a[j];
      a[j] = swap;
   }
}
```


Insertion Sort: Empirical Analysis

Observation. Number of compares depends on input family.
- Descending: $\sim N^2/2$.
- Random: $\sim N^2/4$.
- Ascending: $\sim N$.



Insertion Sort: Mathematical Analysis

Worst case. [descending]
- Iteration i requires i comparisons.
- Total = $(0 + 1 + 2 + ... + N\text{-}1) \sim N^2/2$ compares.

E	F	G	H	I	J	D	C	B	A
						i			

Average case. [random]
- Iteration i requires $i/2$ comparisons on average.
- Total = $(0 + 1 + 2 + ... + N\text{-}1)/2 \sim N^2/4$ compares

A	C	D	F	H	J	E	B	I	G
						i			

Sorting Challenge 1

Q. A credit card company sorts 10 million customer account numbers, for use with binary search.

Using insertion sort, what kind of computer is needed?

A. Toaster
B. Cell phone
C. Your laptop
D. Supercomputer
E. Google server farm

Insertion Sort: Lesson

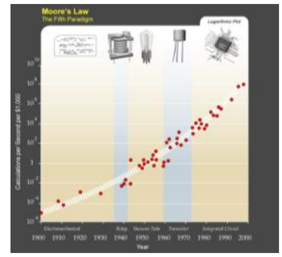
Lesson. Supercomputer can't rescue a bad algorithm.

Computer	Comparisons Per Second	Thousand	Million	Billion
laptop	10^7	instant	1 day	3 centuries
super	10^{12}	instant	1 second	2 weeks

Moore's Law

Moore's law. Transistor density on a chip doubles every 2 years.

Variants. Memory, disk space, bandwidth, computing power per $.



http://en.wikipedia.org/wiki/Moore's_law

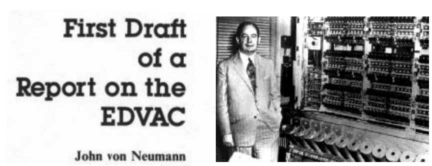
Moore's Law and Algorithms

Quadratic algorithms do not scale with technology.
- New computer may be 10x as fast.
- But, has 10x as much memory so problem may be 10x bigger.
- With quadratic algorithm, takes 10x as long!

"Software inefficiency can always outpace Moore's Law. Moore's Law isn't a match for our bad coding." – Jaron Lanier

Lesson. Need linear (or linearithmic) algorithm to keep pace with Moore's law.

Mergesort



Mergesort

Mergesort algorithm.
- Divide array into two halves.
- Recursively sort each half.
- Merge two halves to make sorted whole.

input
was had him and you his the but

sort left
and had him was you his the but

sort right
and had him was but his the you

merge
and but had him his the was you

Mergesort: Example

M	E	R	G	E	S	O	R	T	E	X	A	M	P	L	E
E	M	R	G	E	S	O	R	T	E	X	A	M	P	L	E
E	M	G	R	E	S	O	R	T	E	X	A	M	P	L	E
E	G	M	R	E	S	O	R	E	T	A	X	M	P	E	L
E	M	G	R	E	S	O	R	T	E	X	A	M	P	L	E
E	M	G	R	E	S	O	R	T	E	X	A	M	P	L	E
E	G	M	R	E	O	R	S	E	T	A	X	M	P	E	L
E	E	G	M	O	R	R	S	A	E	T	X	E	L	M	P
E	M	G	R	E	S	O	R	E	T	X	A	M	P	L	E
E	M	G	R	E	S	O	R	E	T	A	X	M	P	L	E
E	G	M	R	E	O	R	S	A	E	T	X	M	P	E	L
E	M	G	R	E	S	O	R	E	T	A	X	M	P	L	E
E	M	G	R	E	S	O	R	E	T	A	X	M	P	E	L
E	G	M	R	E	O	R	S	A	E	T	X	E	L	M	P
E	E	G	M	O	R	R	S	A	E	E	L	M	P	T	X
A	E	E	E	E	G	L	M	M	O	P	R	R	S	T	X

Merging

Merging. Combine two pre-sorted lists into a sorted whole.

How to merge efficiently? Use an auxiliary array.

i	j	k	aux[k]	a 0	1	2	3	4	5	6	7
				and	had	him	was	but	his	the	you
0	4	0	and	and	had	him	was	but	his	the	you
1	4	1	but	and	had	him	was	but	his	the	you
1	5	2	had	and	had	him	was	but	his	the	you
2	5	3	him	and	had	him	was	but	his	the	you
3	5	4	his	and	had	him	was	but	his	the	you
3	6	5	the	and	had	him	was	but	his	the	you
3	6	6	was	and	had	him	was	but	his	the	you
4	7	7	you	and	had	him	was	but	his	the	you

Trace of the merge of the sorted left half with the sorted right half

Merging

Merging. Combine two pre-sorted lists into a sorted whole.

How to merge efficiently? Use an auxiliary array.

```
int[] aux = new int[N];
// merge into auxiliary array
int i = lo, j = mid;
for (int k = 0; k < N; k++) {
   if      (i == mid) aux[k] = a[j++];
   else if (j == hi)  aux[k] = a[i++];
   else if (a[j] < a[i]) aux[k] = a[j++];
   else                  aux[k] = a[i++];
}

// copy back
for (int k = 0; k < N; k++) {
   a[lo + k] = aux[k];
}
```

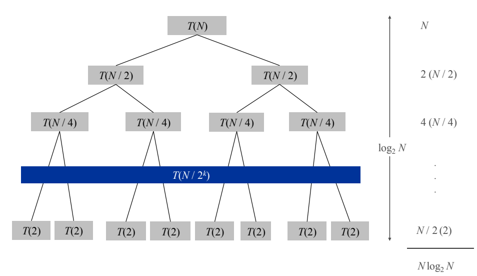

Mergesort: Java Implementation

```
public class Merge {

   public static void sort(int[] a) {
      sort(a, 0, a.length);
   }

   // Sort a[lo, hi).
   public static void sort(int[] a, int lo, int hi) {
      int N = hi - lo;
      if (N <= 1) return;

      // recursively sort left and right halves
      int mid = lo + N/2;
      sort(a, lo, mid);
      sort(a, mid, hi);

      // merge sorted halves (see previous slide)
   }

}
```

Mergesort: Mathematical Analysis

Analysis. To mergesort array of size N, mergesort two subarrays of size $N / 2$, and merge them together using $\leq N$ compares.

we assume N is a power of 2



Mergesort: Mathematical Analysis

Mathematical analysis.

analysis	comparisons
worst	$N \log_2 N$
average	$N \log_2 N$
best	$1/2\ N \log_2 N$

Validation. Theory agrees with observations.

N	actual	predicted
10,000	120 thousand	133 thousand
20 million	460 million	485 million
50 million	1,216 million	1,279 million

Sorting Challenge 2

Q. A credit card company sorts 10 million customer account numbers, for use with binary search.

Using mergesort, what kind of computer is needed?

A. Toaster
B. Cell phone
C. Your laptop
D. Supercomputer
E. Google server farm

Sorting Challenge 3

Q. What's the fastest way to sort 1 million 32-bit integers?

Mergesort: Lesson

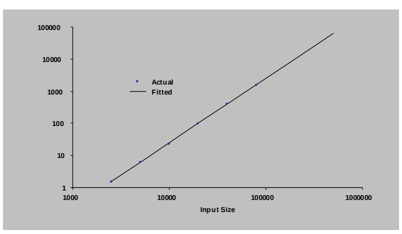
Lesson. Great algorithms can be more powerful than supercomputers.

Computer	Compares Per Second	Insertion	Mergesort
laptop	10^7	3 centuries	3 hours
super	10^{12}	2 weeks	instant

N = 1 billion

Insertion Sort: Empirical Analysis

Data analysis. Plot # comparisons vs. input size on log-log scale.



slope

Hypothesis. # comparisons grows quadratically with input size $\sim N^2 / 4$.

Insertion Sort: Observation

Observe and tabulate running time for various values of N.

- Data source: N random numbers between 0 and 1.
- Machine: Apple G5 1.8GHz with 1.5GB memory running OS X.
- Timing: Skagen wristwatch.

N	Comparisons	Time
5,000	6.2 million	0.13 seconds
10,000	25 million	0.43 seconds
20,000	99 million	1.5 seconds
40,000	400 million	5.6 seconds
80,000	1600 million	23 seconds

Insertion Sort: Prediction and Verification

Experimental hypothesis. # comparisons $\sim N^2/4$.

Prediction. 400 million comparisons for N = 40,000.

Observations.

N	Comparisons	Time
40,000	401.3 million	5.595 sec
40,000	399.7 million	5.573 sec
40,000	401.6 million	5.648 sec
40,000	400.0 million	5.632 sec

Agrees.

Prediction. 10 billion comparisons for N = 200,000.

Observation.

N	Comparisons	Time
200,000	9.997 billion	145 seconds

Agrees.

Insertion Sort: Mathematical Analysis

Mathematical analysis.

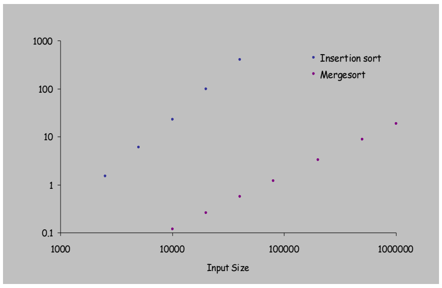
Analysis	Comparisons	Stddev
Worst	$N^2 / 2$	-
Average	$N^2 / 4$	$1/6\, N^{3/2}$
Best	N	-

Validation. Theory agrees with observations.

N	Actual	Predicted
40,000	401.3 million	400 million
200,000	9.9997 billion	10.000 billion

Mergesort: Preliminary Hypothesis

Experimental hypothesis. Number of comparisons ~ 20N.



53

Mergesort: Prediction and Verification

Experimental hypothesis. Number of comparisons ~ 20N.

Prediction. 80 million comparisons for N = 4 million.

Observations.

N	Comparisons	Time
4 million	82.7 million	3.13 sec
4 million	82.7 million	3.25 sec
4 million	82.7 million	3.22 sec

Agrees.

Prediction. 400 million comparisons for N = 20 million.

Observations.

N	Comparisons	Time
20 million	460 million	17.5 sec
50 million	1216 million	45.9 sec

Not quite.

54