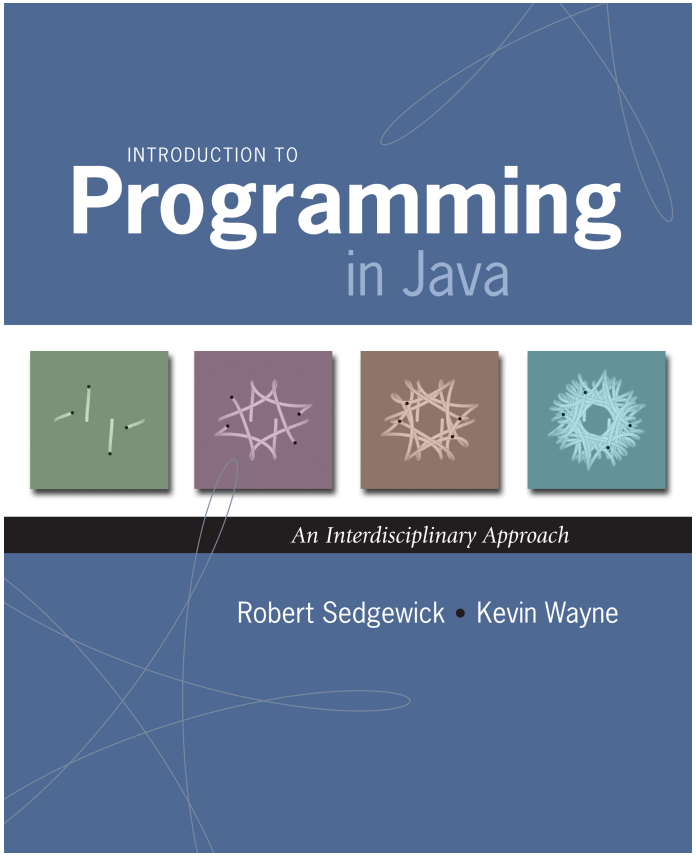
4.2 Mergesort

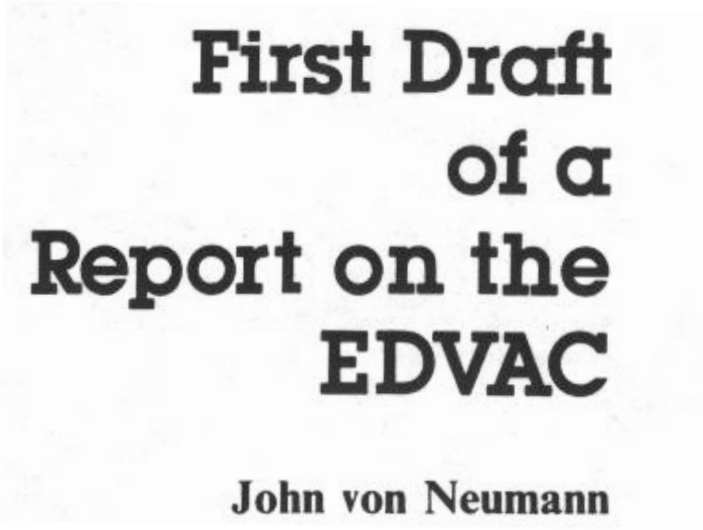


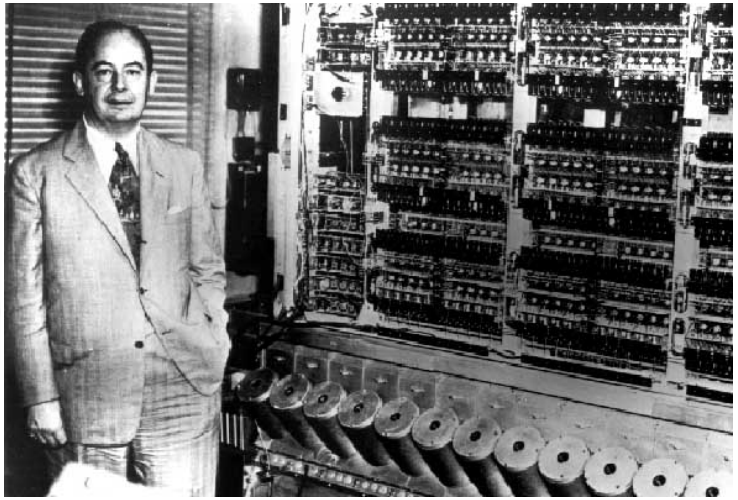
CIS 11 WE HAVE .CLASS

Penn Engineering

Section 4.2

Mergesort





Penn Engineering

Section 4.2

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
```

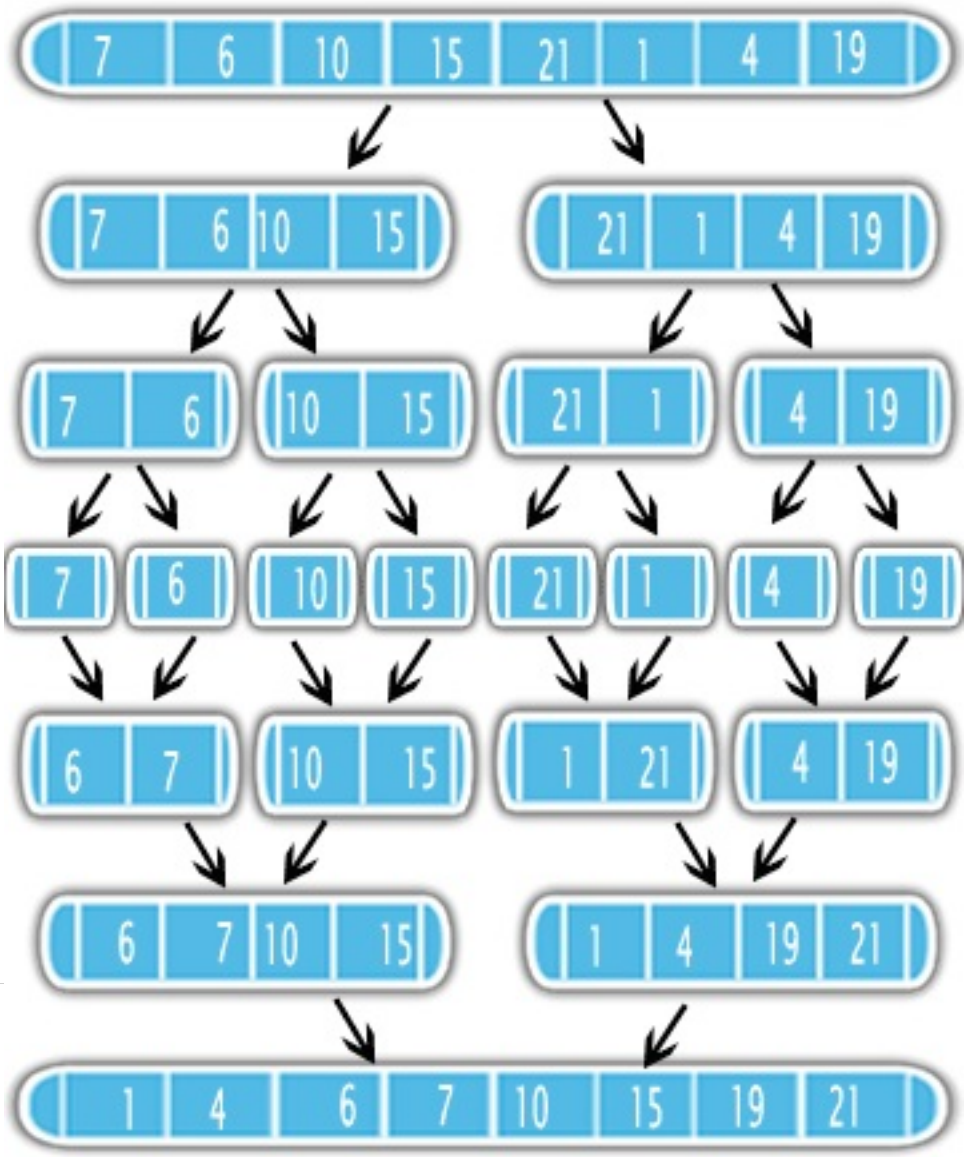
merge
```
and but had him his the was you
```

CIS 11 We Have .CLASS

Penn Engineering

Section 4.2

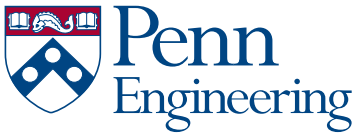
Mergesort: Breakdown



Mergesort: Example

a[]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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	T	E	X	A	M	P	L	E
E	G	M	R	E	S	O	R	T	E	X	A	M	P	L	E
E	G	M	R	E	S	O	R	T	E	X	A	M	P	L	E
E	G	M	R	E	O	R	S	T	E	X	A	M	P	L	E
E	E	G	M	O	R	R	S	T	E	X	A	M	P	L	E
E	E	G	M	O	R	R	S	E	T	X	A	M	P	L	E
E	E	G	M	O	R	R	S	E	T	A	X	M	P	L	E
E	E	G	M	O	R	R	S	A	E	T	X	M	P	L	E
E	E	G	M	O	R	R	S	A	E	T	X	M	P	L	E
E	E	G	M	O	R	R	S	A	E	T	X	M	P	E	L
E	E	G	M	O	R	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



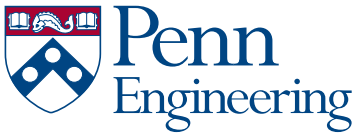
Section 4.2

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



Section 4.2

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

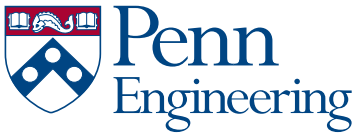
```
public class Merge {

   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)
   }
}
```

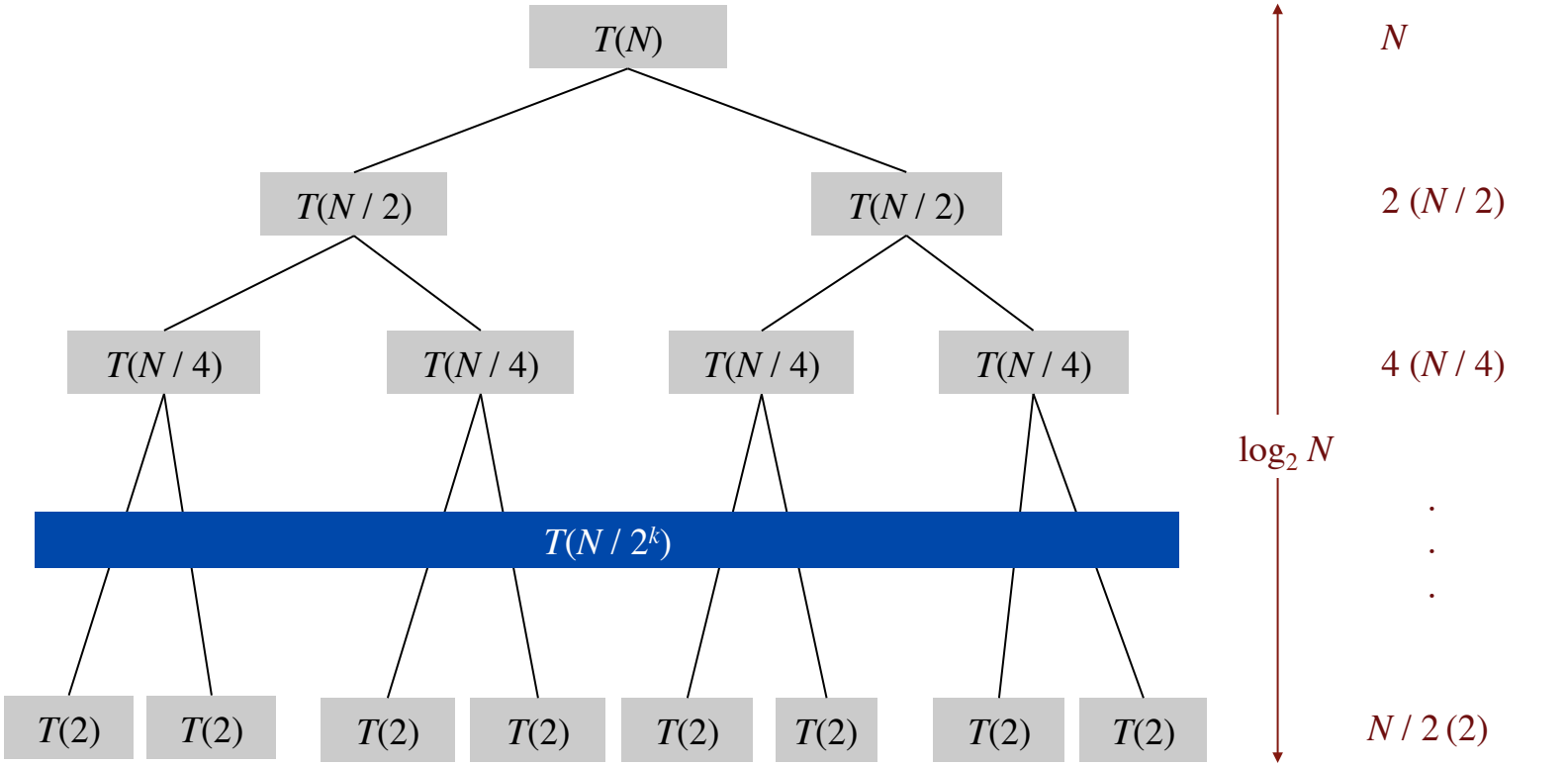


Section 4.2

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



CIS 11 We Have .CLASS

Penn Engineering

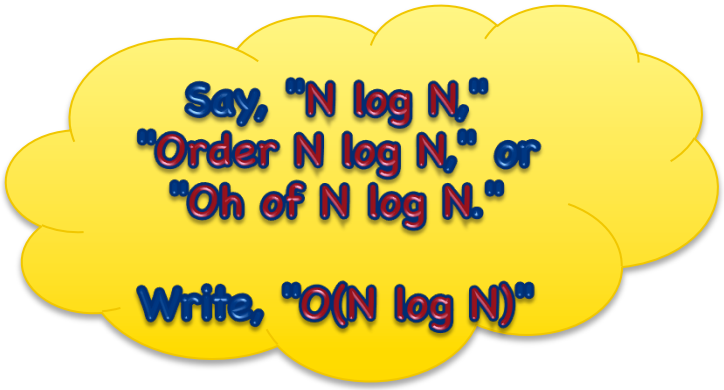
Programming in Java

Section 4.2

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.

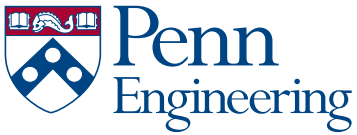
N	actual	predicted
10,000	120 thousand	133 thousand
20 million	460 million	485 million
50 million	1,216 million	1,279 million

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



Section 4.2

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