INTRODUCTION TO COMPUTER SCIENCE

Rutgers University

14. Recursive Search and Sort

http://introcs.cs.rutgers.edu

INTRODUCTION TO COMPUTER SCIENCE

Rutgers University

14. Recursive search and sort

- Recursive binary search
- Performance analysis
- Mergesort
- Mergesort analysis

Styles Adopted from Slides by Sedgewick and Wayne

Recursion trace for binary search

Given an ordered array of values, write a recursive binary search method to locate a specified value in the array. If the value is not present in the array, indicate that.
Identify the base case and the general case in the algorithm. array of values are provided.

LO 14.1a

```
public static int search(String key, String[] a)
{  return search(key, a, 0, a.length); }

public static int search(String key, String[] a,
                         int lo, int hi)
{
   if (hi <= lo) return -1;
   int mid = lo + (hi - lo) / 2;
   int cmp = a[mid].compareTo(key);
   if      (cmp > 0) return search(key, a, lo, mid);
   else if (cmp < 0) return search(key, a, mid+1, hi);
   else              return mid;
}
```

```
search("oscar")
   return search(... 10);

   search("oscar", a, 0, 15)
      mid = 7;
      > "eve"
      return search(... 10);

   search("oscar", a, 8, 15)
      mid = 11;
      < "peggy"
      return search(... 10);

   search("oscar", a, 8, 11)
      mid = 9;
      > "mallory"
      return search(... 10);

   search("oscar", a, 10, 11)
      mid = 10;
      == "oscar"
      return 10;
```

14. Recursive search and sort

- Recursive binary search
- Performance analysis
- Mergesort
- Mergesort analysis

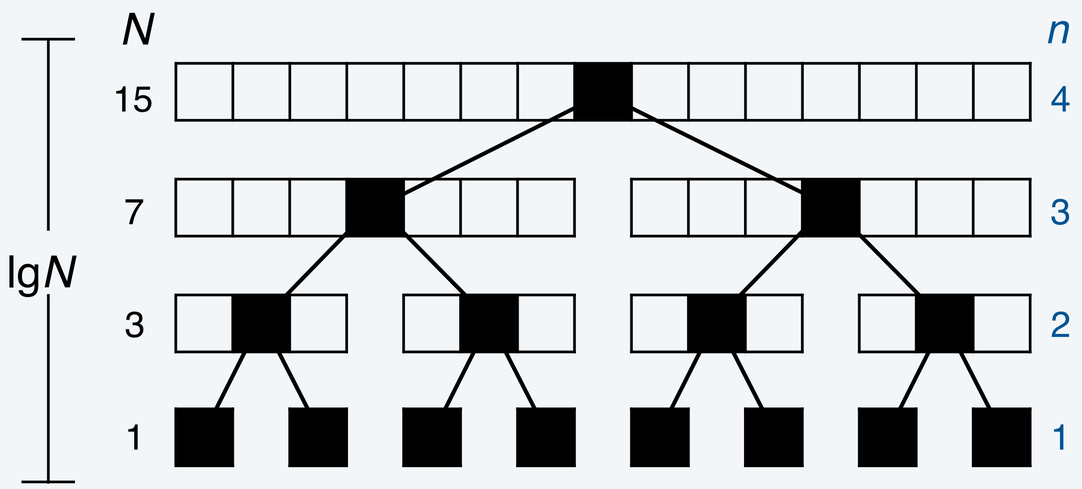
Mathematical analysis of binary search

Count the number of comparisons made when a recursive binary search algorithm is executed, and the search key and the array of values are provided.

LO 14.1b

Exact analysis for search miss for $N = 2^n - 1$

- Note that $n = \lg(N+1) \sim \lg N$.
- Subarray size for 1st call is $2^n - 1$.
- Subarray size for 2nd call is $2^{n-1} - 1$.
- Subarray size for 3rd call is $2^{n-2} - 1$.
- ...
- Subarray size for nth call is 1.
- Total # compares (one per call): $n \sim \lg N$.



Every search miss is a top-to-bottom path in this tree.

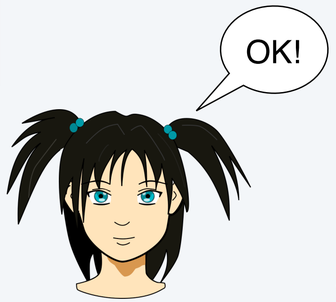
Proposition. Binary search uses ~lg N compares for a search miss.

Proof. An (easy) exercise in discrete math.

Proposition. Binary search uses ~lg N compares for a random search hit.

Proof. A slightly more difficult exercise in discrete math.

Interested in details? Take a course in algorithms.



Empirical tests of binary search

Whitelist filter scenario

- Whitelist of size *N*.
- 10*N* transactions.

N	T_N (seconds)	$T_N/T_{N/2}$	*transactions per second*
100,000	1		
200,000	3		
400,000	6	2	67,000
800,000	14	2.35	57,000
1,600,000	33	2.33	48,000
10.28 million	264	2	48,000

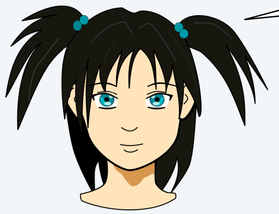
```
% java Generator 100000 ...
1 seconds
% java Generator 200000 ...
3 seconds
% java Generator 400000 ...
6 seconds
% java Generator 800000 ...
14 seconds
% java Generator 1600000 ...
33 seconds
```

... = 10 a-z l java TestBS
a-z = abcdefghijklmnopqrstuvwxyz

nearly 50,000 transactions per second, and holding

Validates hypothesis that order of growth is *N* log *N*.

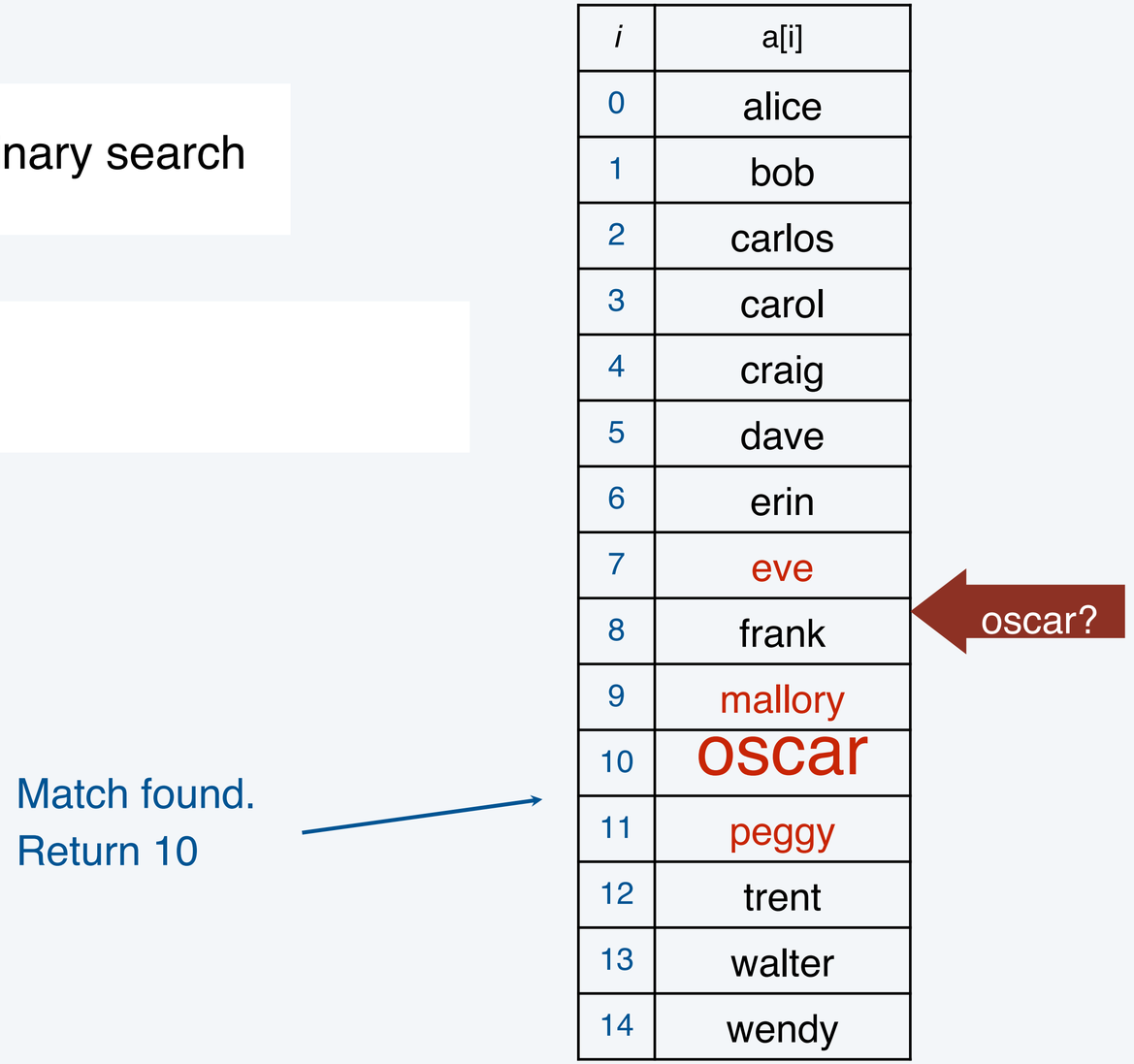
Will scale.



Understanding recursive binary search

Identify. All elements that are inspected during binary search

comparisons. 4 names inspected



Best, worst and average case of recursive binary search

Determine the best case, average case, and worst case Big-O analysis for a recursive binary search algorithm when the search key and the array of values are provided.

LO 14.1d

Best case. Oscar is the first record (log N comparison)

Worst case. Oscar is the last record (log N comparisons)

Average case. Oscar is the “middle” record (log N comparisons)

i	a[i]
0	alice
1	bob
2	carlos
3	carol
4	craig
5	dave
6	erin
7	eve
8	frank
9	mallory
10	oscar
11	peggy
12	trent
13	walter
14	wendy

14. Recursive search and sort

- Recursive binary search
- Performance analysis
- **Mergesort**
- Mergesort analysis

Mergesort algorithm

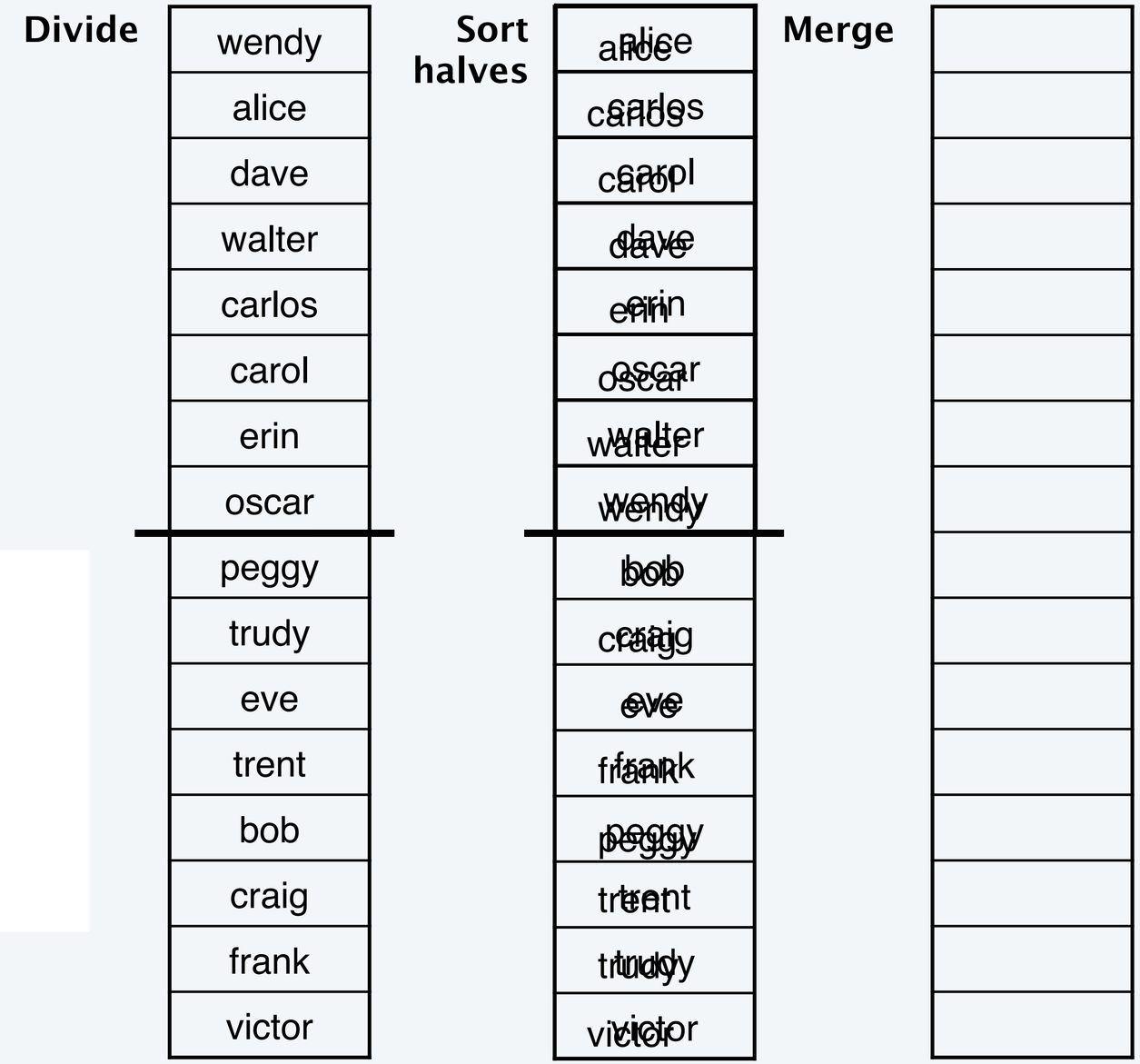
Mergesort

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

John von Neumann
1903–1957

John von Neumann

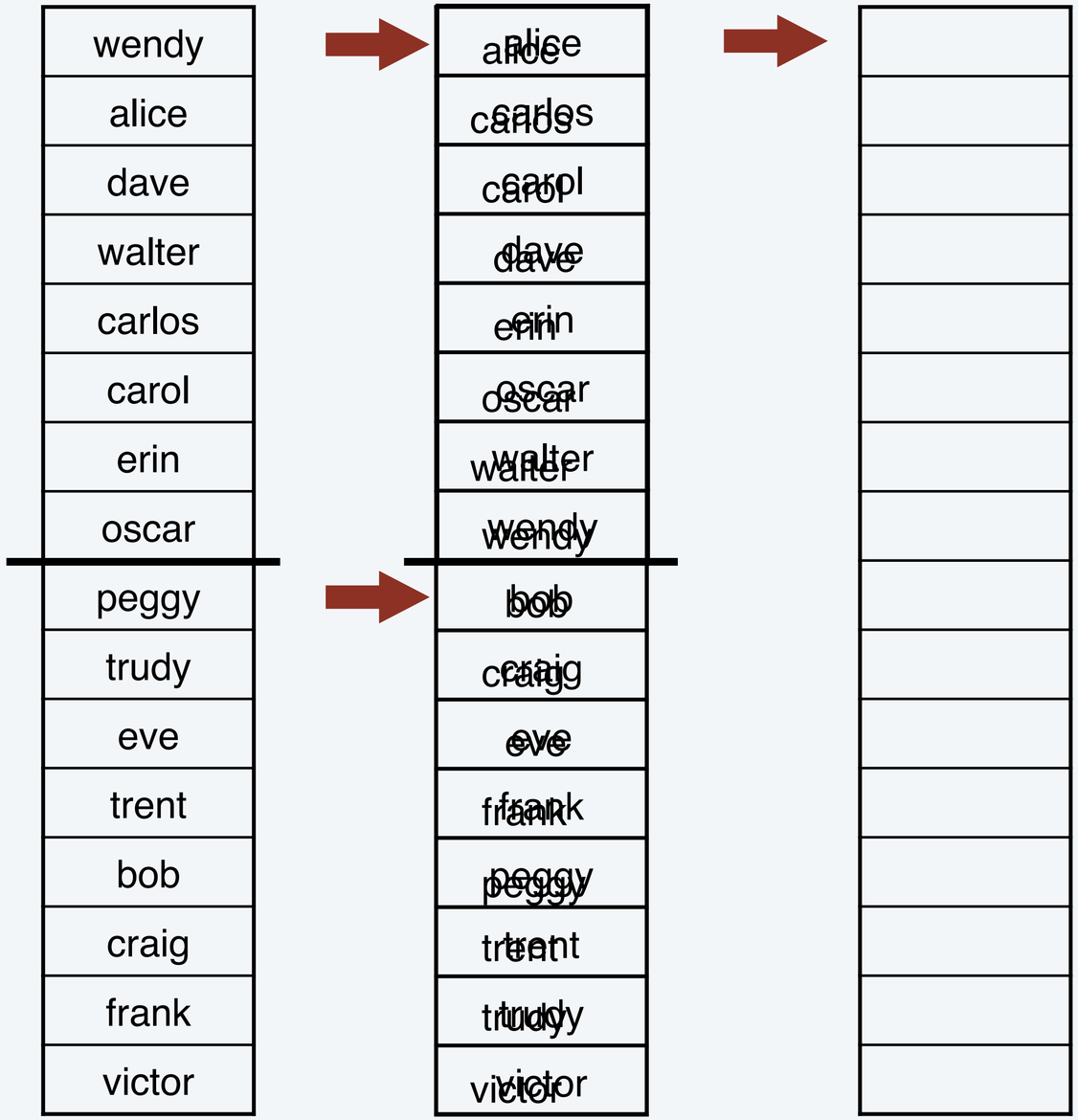
- Pioneered computing (stay tuned).
- Early focus on numerical calculations.
- Invented mergesort as a test to see how his machine would measure up on other tasks.



Mergesort algorithm

Mergesort

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



Merge: Java implementation

Merge

- Merge a[lo, mid] with a(mid, hi].

```
public static void merge (String[] a, int lo, int m, int hi){
  //copy lower half of the array into b
  String[] b = new String[m-lo+1];
  for(int i=0;i<=m-lo;i++) { b[i] = a[lo+i]; }
  int i=0,j=m+1,k=lo;
  while(i <= m-lo && j <= hi) {
    if (a[j].compareTo(b[i]) < 0) {
      // a[j] < b[i]
      a[k] = a[j]; k+=1; j+=1;
    } else {
      a[k] = b[i]; k+=1; i+=1;
    }
  }
  // copy remaining
  while ( i <= m-lo ) { a[k] = b[i]; k+=1; i+=1; }
  while ( j <= hi ) { a[k] = a[j]; k+=1; j+=1;}
```

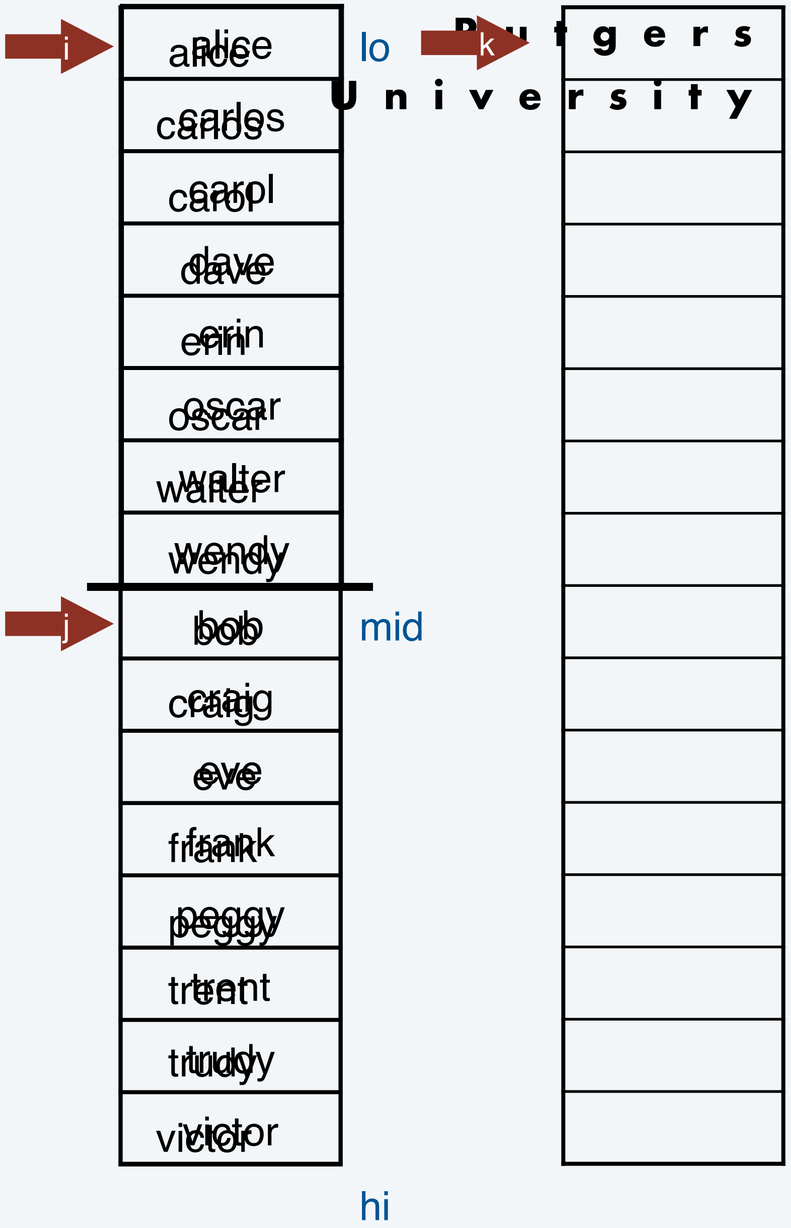
Merge: animation

Given an array of values, give a step-by-step illustration of executing the ***Mergesort*** on the array. State the array contents after each pass of the sort

LO 14.2a

INTRODUCTION TO COMPUTER SCIENCE

Rutgers University



Mergesort: Java implementation

Mergesort

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

```
% more names16.txt
wendy
alice
dave
walter
carlos
carol
erin
oscar
peggy
trudy
eve
trent
bob
craig
frank
victor
```

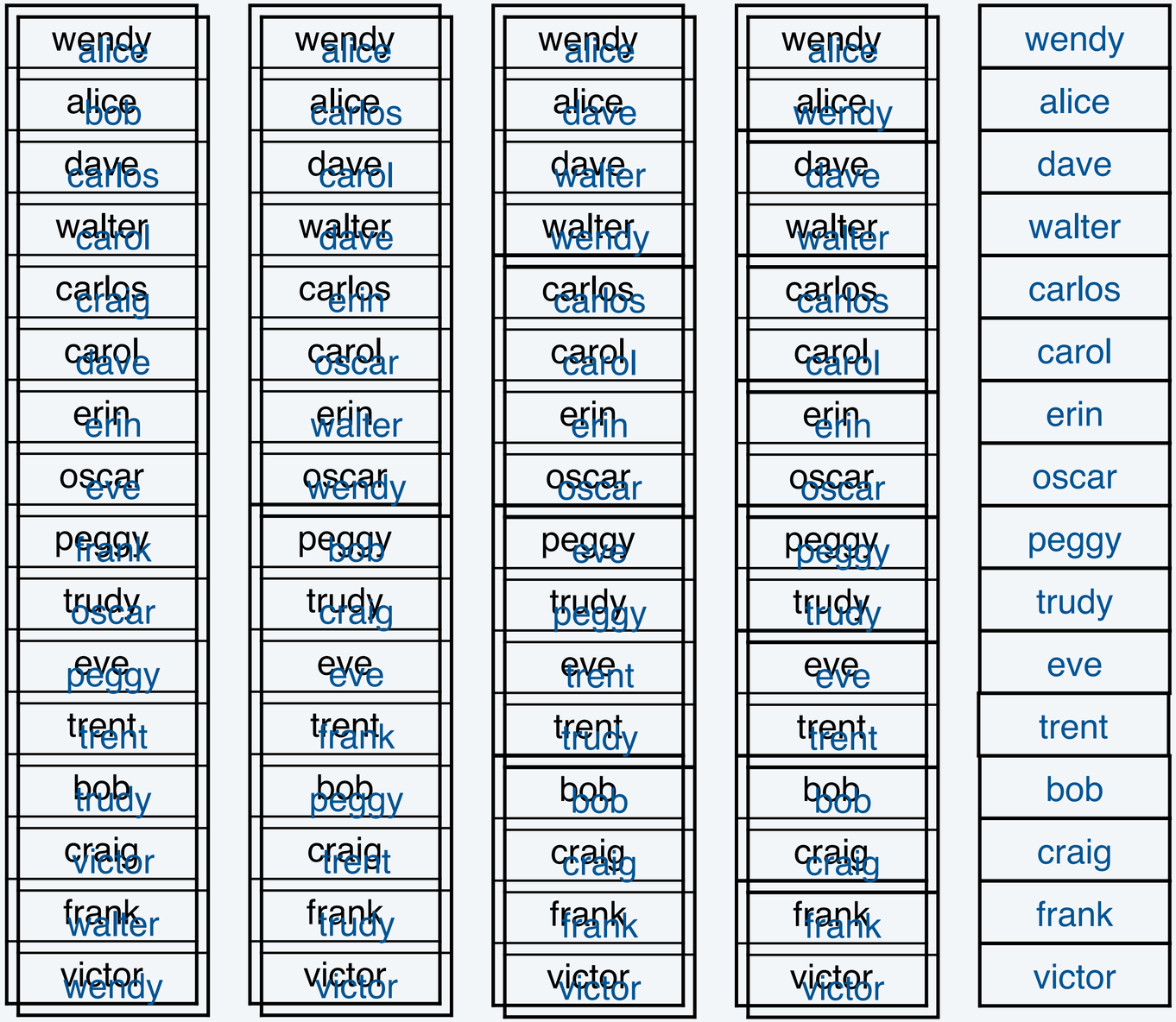
```
% java Merge < names16.txt
alice
bob
carlos
carol
craig
dave
erin
eve
frank
oscar
peggy
trent
trudy
victor
walter
wendy
```

```
public class Merge
{
   private static String[] aux;

   public static void merge(String[] a, int lo, int mid, int hi)

   { // See previous slide.  }

   public static void sort(String[] a)
   {
      aux = new String[a.length];  // Allocate just once!
      sort(a, 0, a.length);
   }


public static void sort (String[] a, int lo, int hi) {

  if (lo >= hi) return;

  int middle = (lo+hi)/2;

  mergesort(a, lo, middle);

  mergesort(a, middle+1, hi);

  merge(a, l, middle, r);

}
```

Mergesort trace - animation

Mergesort

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



14. Recursive search and sort

- Recursive binary search
- Performance analysis
- Mergesort
- Mergesort analysis

Mergesort analysis

Determine the best case, average case, and worst case Big-O analysis of the mergesort.

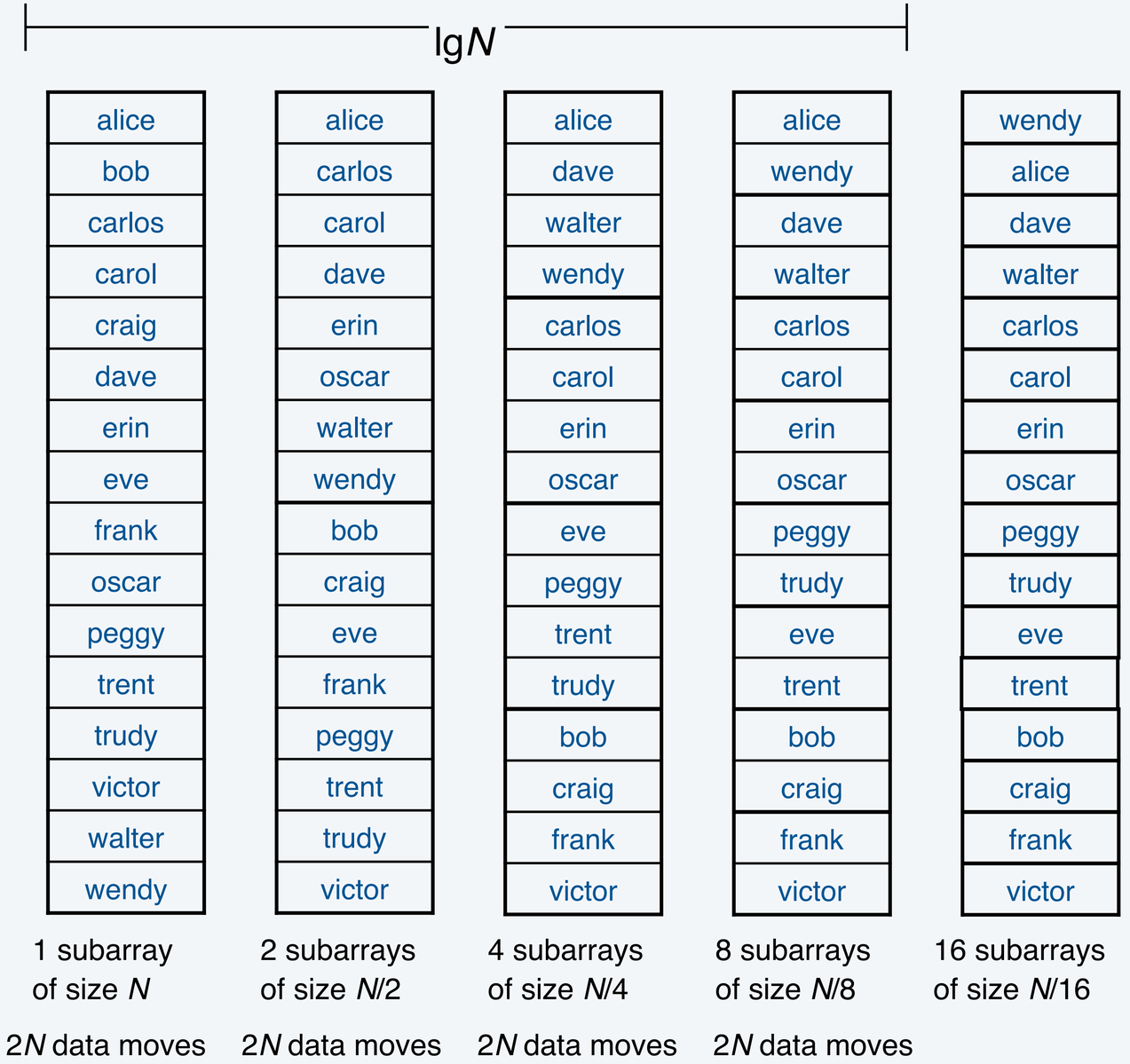
Cost model. Count *data moves.*

# of times a string moves from one array to another

Exact analysis for $N = 2^n$.

- Note that $n = \lg N$.
- 1 subarray of size 2^n.
- 2 subarrays of size 2^{n-1}.
- 4 subarrays of size 2^{n-2}.
- ...
- 2^n subarrays of size 1.
- Total # data moves: $2N \lg N$.

Interested in details?
Take a course in algorithms.



Empirical tests of mergesort

Sort random strings

- Array of length *N*.
- 10-character strings.

N	T_N (seconds)	$T_N/T_{N/2}$
1 million	1	
2 million	2	
4 million	5	2.5
8 million	10	2
16 million	20	2.5
...		
1.02 billion	1280	2

20 minutes

```
% java Generator 1000000 ...
1 seconds
% java Generator 2000000 ...
2 seconds
% java Generator 4000000 ...
5 seconds
% java Generator 8000000 ...
10 seconds
% java Generator 16000000 ...
20 seconds
```

... = 10 a-z | java Merge
a-z = abcdefghijklmnopqrstuvwxyz

Confirms hypothesis that order of growth is *N* log *N*

WILL scale

INTRODUCTION TO COMPUTER SCIENCE

Rutgers University

14. Recursive search and sort

- Recursive binary search
- Performance analysis
- Mergesort
- Mergesort analysis

INTRODUCTION TO COMPUTER SCIENCE

Rutgers University

14. Recursive Search and Sort

http://introcs.cs.rutgers.edu