CSE 12 – Basic Data Structures

Prof. Christine Alvarado
[Slides borrowed/adapted from slides by Cynthia Lee, Rakesh Varna, & Roshni Chandrashekhar]
Announcements

1. Midterms can be picked up in discussion this week
   1. Most people did very well. Grades will be on Gradesource later today
2. Thank you for your feedback. Still reading it, but stay tuned for my response(s)
3. NO CLASS on Wednesday, BUT:
   1. There IS a QuizStar quiz due by class time
   2. There will be a video lecture. I expect you to watch it, and DO the problems just like class
Review Quiz Problem 2

- Answer D should be “At most N-1 but usually less than N-1”
Today’s topics

- Quicksort
- More with Java Generics
public void mergeSort( int[] toSort, int start, int end )
{
    if ( start < end ) {
        int mid = start + ((end - start) / 2);
        mergeSort( toSort, workArray, start, mid );
        mergeSort( toSort, workArray, mid+1, end );
        merge( toSort, workArray, start, mid, end );
    }
}

This mergeSort works!!
Merge Sort follows a series of splitting steps with a series of merging steps to produce sorted partitions.
Merge Sort: An Example

How many split steps are there?
A. \( \log(N) \)
B. \( N \)
C. \( N^2 \)

Merge Sort follows a series of splitting steps with a series of merging steps to produce sorted partitions.
About how many instructions are executed at each split?
A. 1
B. $\log(N)$
C. $N$
D. $N^2$

Merge Sort follows a series of splitting steps with a series of merging steps to produce sorted partitions.
Merge Sort: An Example

How many merge steps are there?
A. $\log(N)$
B. $N$
C. $N^2$

Merge Sort follows a series of splitting steps with a series of merging steps to produce sorted partitions.
Merge Sort: An Example

How many comparisons are made at each merge? (total)

A. \( \log(N) \)
B. \( N \)
C. \( N^2 \)
D. Other

Merge Sort follows a series of splitting steps with a series of merging steps to produce sorted partitions.
Running time for Mergesort:
\[ \log(N) \times N + \log(N) \times N = O(N \times \log(N)) \]
Quicksort: Another magical (recursive) algorithm

| 12 | 4 | 9 | 14 | 15 | 8 | 19 | 2 |

Select a **pivot** element:

| 12 | 4 | 9 | 14 | 15 | 8 | 19 | 2 |

“Partition” the elements in the array (smaller, pivot, larger)

| 8 | 4 | 9 | 2 | 12 | 15 | 19 | 14 |

Magically sort the smaller elements and the larger elements (Quicksort)

| 2 | 4 | 8 | 9 | 12 | 15 | 19 | 21 |
Quicksort: Another magical (recursive) algorithm

Select a **pivot** element:

```
  12  4  9  14  15  8  19  2
```

“Partition” the elements in the array (**smaller**, **pivot**, **larger**)

```
  8  4  9  2  12  15  19  14
```

We won’t cover how the partition step works, but see if you can figure it out! (or Google it)
Quick Sort: Using a “good” pivot

How many levels will there be if you choose a pivot that divides the list in half?

A. 1
B. \( \log(N) \)
C. \( N \)
D. \( N\log(N) \)
E. \( N^2 \)
Quick Sort: Using a “good” pivot

If the time to partition on each level takes N comparisons, how long does Quicksort take with a good partition?

A. O(1)
B. O(log(N))
C. O(N)
D. O(N*log(N))
E. O(N^2)
Quick Sort: Using a “good” pivot

If the time to partition on each level takes $N$ comparisons, how long does Quicksort take with a good partition?

A. $O(1)$
B. $O(\log(N))$
C. $O(N)$
D. $O(N\log(N))$
E. $O(N^2)$

Space complexity: $O(\log_2 n)$ for the runtime stack activation records
Which of these choices would be the worst choice for the pivot?

A. The minimum element in the list
B. The last element in the list
C. The first element in the list
D. A random element in the list
Quick sort with a bad pivot

If the pivot always produces one empty partition and one with $n - 1$ elements, there will be $n$ levels, each of which requires $O(n)$ comparisons: $O(n^2)$ time complexity

Space complexity: $O(n)$ for the activation records
Which of these choices is a better choice for the pivot?

A. The first element in the list
B. A random element in the list
C. They are about the same