Sorting

Discussion 12

Announcements

- Lab 12 due Monday 04/11
- Week 12 survey due Tuesday 4/12
- HW 7 due Tuesday 04/12
- Test 2 Regrades close Friday 04/15
- Lab 13 due Friday 04/15
- Project 3 Released!

Review

Insertion Sort

Insertion sort takes each item from the list and inserts it into its correct place by swapping backwards.

3 5 1 2 4

Runtime: $O(N^2)$

Selection Sort

Selection sort finds the smallest remaining element in the unsorted portion of the list at each time step.

3 5 1 2 4

Runtime: $\Theta(N^2)$

Merge Sort

Merge sort splits the list in half, applies merge sort to each half, and then merges the two halves together in a zipper fashion.

3 5 1 2 4

Runtime: Θ(NlogN)

Heap Sort

Heapsort heapifies the array into a max heap and pops the largest element off and appends it to the end until there are no elements left in the heap.

3 5 1 2 4

Runtime: O(NlogN)

Quicksort

Quicksort picks a partition and uses Hoare's partition to divide the list so that everything greater than the partition is on its right and everything less than the partition is on its left.

3 5 1 2 4

Runtime: $\Theta(N \log N)^*$ (Amortized - worst case: $\Theta(N^2)$)

LSD Radix Sort

LSD sorts numbers by sorting them by digit from lowest digit to largest digit.

```
120
923
112
342
199
```

Runtime: $\Theta(W(N + R))$

MSD Radix Sort

MSD sorts numbers by sorting them by digit from largest digit to smallest digit..

```
120
923
112
342
199
```

Runtime: O(W(N + R))

Worksheet

1A Sorting: Mechanical Practice

Show the steps taken by Quicksort on the following unordered list.

`106, 351, 214, 873, 615, 172, 333, 564`

1B Sorting: Mechanical Practice

Show the steps taken by Merge Sort on the following unordered list.

`106, 351, 214, 873, 615, 172, 333, 564`

1C Sorting: Mechanical Practice

Show the steps taken by LSD Radix Sort on the following unordered list.

`106, 351, 214, 873, 615, 172, 333, 564`

2A Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
12, 7, 8, 4, 10, 2, 5, 34, 14

7, 8, 4, 10, 2, 5, 12, 34, 14

4, 2, 5, 7, 8, 10, 12, 14, 34
```

2A Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
12, 7, 8, 4, 10, 2, 5, 34, 14

7, 8, 4, 10, 2, 5, 12, 34, 14

4, 2, 5, 7, 8, 10, 12, 14, 34
```

Quicksort, using the first element as a pivot.

2B Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
23, 45, 12, 4, 65, 34, 20, 43
```

```
4, 12, 23, 45, 65, 34, 20, 43
```

2B Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
23, 45, 12, 4, 65, 34, 20, 43
```

```
4, 12, 23, 45, 65, 34, 20, 43
```

Insertion Sort.

2C Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
12, 32, 14, 11, 17, 38, 23, 34
```

```
12, 14, 11, 17, 23, 32, 38, 34
```

2C Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
12, 32, 14, 11, 17, 38, 23, 34
```

```
12, 14, 11, 17, 23, 32, 38, 34
```

MSD Radix Sort.

2D Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
45, 23, 5, 65, 34, 3, 76, 25

23, 45, 5, 65, 3, 34, 25, 76

5, 23, 45, 65, 3, 25, 34, 76
```

2D Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
45, 23, 5, 65, 34, 3, 76, 25

23, 45, 5, 65, 3, 34, 25, 76

5, 23, 45, 65, 3, 25, 34, 76
```

Merge Sort.

2E Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
23, 44, 12, 11, 54, 33, 1, 41

54, 44, 33, 41, 23, 12, 1, 11

44, 41, 33, 11, 23, 12, 1, 54
```

2E Sorting: Identification

Determine what sorting algorithm results in these intermediate steps.

```
23, 44, 12, 11, 54, 33, 1, 41
```

```
54, 44, 33, 41, 23, 12, 1, 11
```

```
44, 41, 33, 11, 23, 12, 1, 54
```

Heapsort.

3 Conceptual Sorts

A. True or False: Quicksort has a worst case runtime of NlogN, where N is the number of elements in the list we are sorting.

B. We have a system running insertion sort and we find that its completing fasting than expected. What can we conclude about the input?

C. Give a 5 integer array that elicits the worst case runtime for insertion sort.

D. True or False: Heapsort is stable.

E. Why would someone choose mergesort over quicksort?

3 Conceptual Sorts

A. **True or False: Quicksort has a worst case runtime of NlogN, where N is the number of elements in the list we are sorting.**

   False

B. **We have a system running insertion sort and we find that its completing faster than expected. What can we conclude about the input?**

   The input is small or the array is nearly sorted.

C. **Give a 5 integer array that elicits the worst case runtime for insertion sort.**

   5 4 3 2 1

D. **True or False: Heapsort is stable.**

   False

E. **Why would someone choose mergesort over quicksort?**

   The worst case runtime of mergesort is Θ(NlogN), while quicksort is Θ(N^2). Additionally, mergesort is stable while quicksort is not.

3 Conceptual Sorts

Which sorts do each of the following statements describe?

Bounded by Ω(NlogN) lower bound:

Has a worst case runtime that is asymptotically better than Quicksort's:

In the worst case, performs Θ(N) pairwise swaps:

Never compares the same two elements twice:

Runs in best case Θ(logN) for certain inputs:

3 Conceptual Sorts

Which sorts do each of the following statements describe?

Bounded by Ω(NlogN) lower bound: A, B, C

Has a worst case runtime that is asymptotically better than Quicksort's: B, E

In the worst case, performs Θ(N) pairwise swaps: C

Never compares the same two elements twice: A, B, D

Runs in best case Θ(logN) for certain inputs: N