Algorithm Course Introduction

Slides adapted from
Brandon Fain

Outline

- Homework
  - LaTeX: How to type mathematics
  - Gradescope
- Math and Programming: What you should already know
- Sorting methods and their recurrence relations
  - Selection sort
  - Insertion sort
  - Merge sort
  - Quick sort

Homework

- Released on Sakai
- Submit on Gradescope
- Label each question when submitting
- Preferably type your homework. Preferably LaTeX but not required.

LaTeX: How to type mathematics

- “It is strongly encouraged that students should type their solutions using LaTeX...If handwritten solutions are illegible, they will not be graded.”
- LaTeX is the preeminent markup language for writing well formatted mathematics.
- Works like html – you write a source .tex document that is mostly text and equations, with commands specified with a ‘/’. The .tex is compiled into a .pdf for viewing.
- You can find free distributions of LaTeX from latex-project.org/get or you can use a free web based editor like overleaf.com.

LaTeX: How to type mathematics

- Let's look at a brief demo LaTeX document.
- This demo document is available under resources on Sakai.
- More resources and references:
  - A short introduction translated into many languages: ctan.org/pkg/lshort.
  - Tutorials for learning LaTeX sharelatex.com/learn.
  - LaTeX cheat sheet (2 page reference): https://wch.github.io/latexsheet/

Outline

- ~~Homework~~
  - ~~LaTeX: How to type mathematics~~
  - ~~Gradescope~~

- Math and ~~Programming~~: What you should already know

- Sorting methods and their recurrence relations
  - Selection sort
  - Insertion sort
  - Merge sort
  - Quick sort

Math: What You Should Already Know

- **Algebra:** logarithms/exponentials, manipulating finite sequences and summations, systems of equations and inequalities.
- **Counting/Probability:** events, inclusion/exclusion, pigeonhole principle, conditional probability, random variables, and expectations.
- **Logic/Proofs:** boolean formulas, quantifiers, set theory, proof by contradiction, weak & strong induction, and writing proofs in English.
- **Linear Algebra:** basic definitions and matrix multiplication.
- **Graph Theory:** basic definitions, degrees, isomorphisms, bipartite graphs, and matchings.
- **More information:** Check out a recent iteration of a CompSci 230. If you need to brush up, Mathematics for Computer Science is a good reference.

Outline

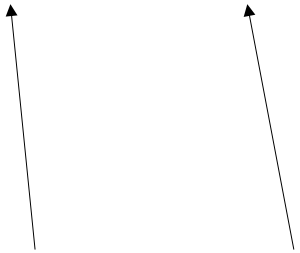
- ~~Homework~~
  - ~~LaTeX: How to type mathematics~~
  - ~~Gradescope~~

- ~~Math and Programming: What you should already know~~

- Sorting methods and their recurrence relations
  - Selection sort
  - Insertion sort
  - Quick sort
  - Merge sort

Sorting and Recurrence Relations

- The goal of a sorting algorithm is to put things in order
- A recurrence relation is some recursive equation that shows operations taken at each iteration
- $T(n) = aT(cn-d) + f(n)$



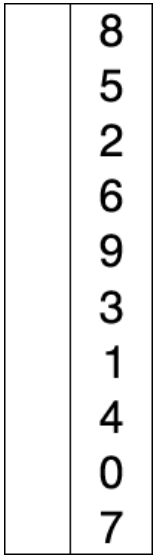
How does the workspace change on the next step

What operations are happening at the current step

Really naïve ways to sort?

Selection sort

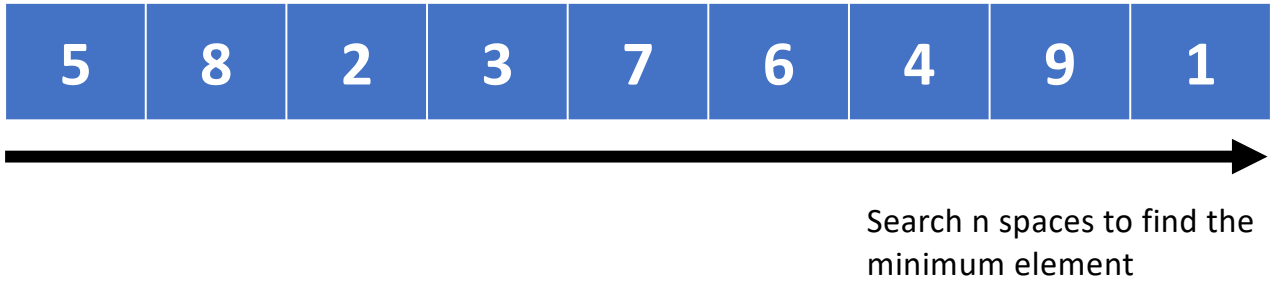
- Sorted elements at the start
- Unsorted next
- Find the minimum element and put it at the end of the sorted list
- Worst, average, best case all $O(n^2)$



From Wikipedia

Recurrence relation for selection sort

- How to build the recurrence relation for selection sort?



Recurrence relation for selection sort

- How to build the recurrence relation for selection sort?

Recursive call is 1 less element

$$T(n) = \underbrace{T(n-1)}_{\text{Recursive call}} + \underbrace{n}_{\text{Search n spaces}}$$

Exercise: Solve the recurrence relation

Insertion Sort

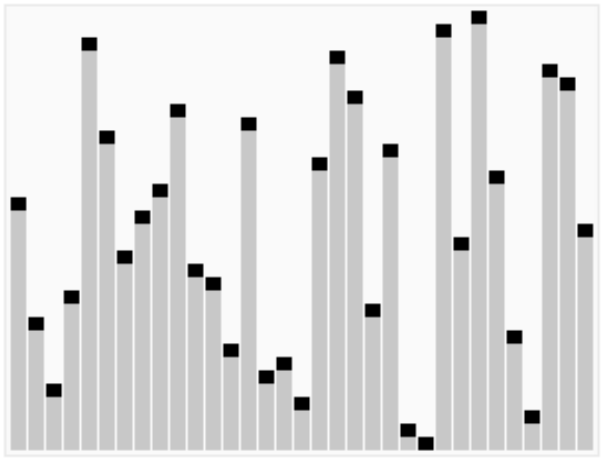
- Make a sorted list
- Visit each unsorted element, and put it in its sorted position in the list
- Best case is $O(n)$
  - What changed compared to selection sort?
- Worst and average are $O(n^2)$
- Still T(n) = T(n-1) + n
  - Can you explain each of the terms w.r.t. insertion sort?

6 5 3 1 8 7 2 4

From Wikipedia

Quick sort

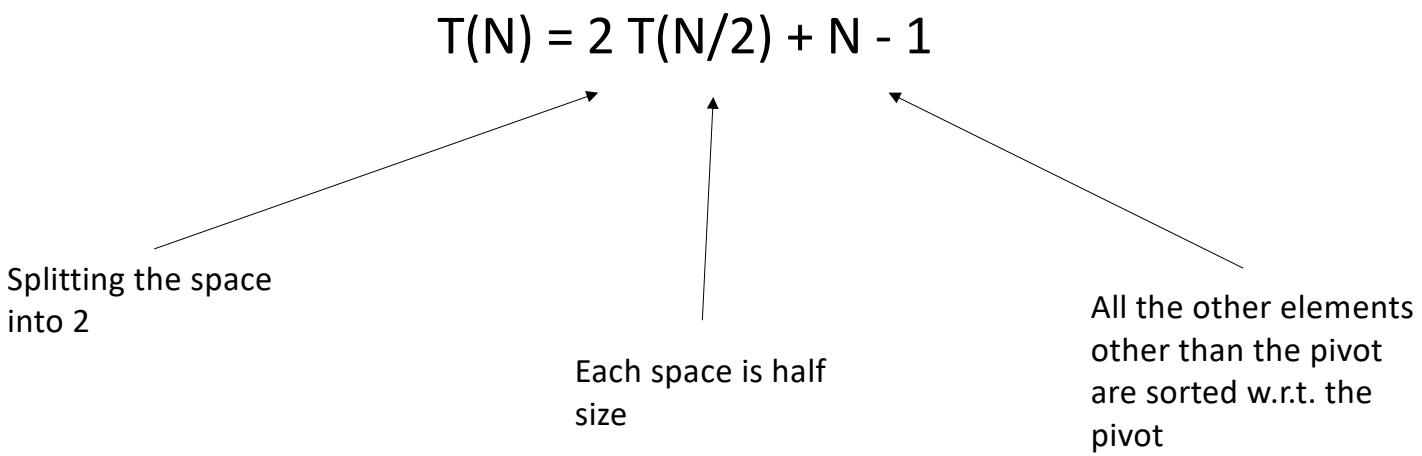
- Partition high or low elements w.r.t a pivot
- Repeat within each partition
- Usually $O(n \log n)$, but can rarely still take $O(n^2)$
  - What is the rare worst case?



From Wikipedia

Recurrence relation for quick sort

- Best case (if median pivot is always chosen)



Recurrence relation for quick sort

- Worst case (if a left or rightmost pivot is selected for a list that is already in order or reverse order)

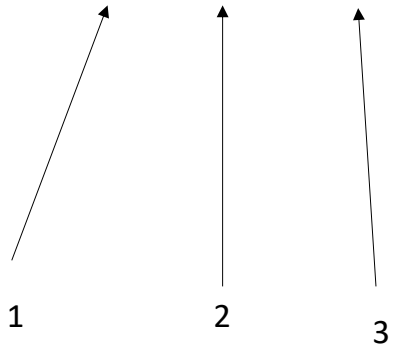
$$T(N) = T(N-1) + N - 1$$

The pivot always ends up on the rightmost element of a list that is 1 element smaller

Check every element except the pivot

Merge sort

- Keep splitting until pairs
- Compare, then merge and repeat
- Always runs in $O(n \log n)$
- T(N) = 2 T(N/2) + N



6 5 3 1 8 7 2 4

From Wikipedia