

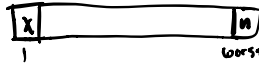
Section 2.5

- * Go back to finish chapter 2. Skip Fib section for now
- * Finish homework #2
- * Test OI on 09/16/2020 → everything through now 2.

• Empirical analysis

- Analyze Best + worst case
easier than average

Seq. Search



$$\frac{1+2+3+\dots+n}{n}$$

- Let's us take an implementation and benchmark it.

↳ Generate range of differing inputs. → Size, complexity, order... etc...

↳ Performance on real hardware.

↳ Collect data on memory use + on exec. time] collect info
gather metrics

• Perturbation: perturbing (altering) what you are observing.

Chapter 3: Brute Force techniques ex. sequential search.

- Exhaustive search → have to look at every value. Special category of Brute Force.

Selection Sort

89 | 45 | 68 | 90 | 29 | 34 | 17

17 | 45 | 68 | 90 | 29 | 34 | 89

17 | 34 | 68 | 90 | 29 | 45 | 89

⋮

Find min and move to front.

Still $(n-1) + (n-2) + \dots + \frac{n(n-1)}{2}$

$\Theta(n^2)$ → comparisons

$\Theta(n)$ → swaps

- Selection will be faster because

there are not as many swaps.

- Only need a temp, so it is in-place.

↳ No duplicate array.

↳ Both are in-place

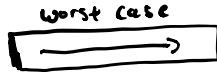
- Stable?

↳ Bubble sort yes

↳ Selection sort no

Searching

Sequential Search



n comparisons

3.3/3.4 preview.

3.3 \rightarrow Closest pair

3.4: Exhaustive Search

- Combinatorial problems
- Often looking for optimal solution.

Knapsack

#1 \$5

#3 \$6

#7 \$1



Max value
w/ weight

Exhaustive
Search

□ △ ○

0	0	0
0	0	1
0	1	0
0	1	1

2^3

8 comparisons

generate sets

...

1	1	1
---	---	---