COP 3503 2/15/2024

Lower bound for comparison sorting

CS1 - Merge Sort O(nlgn)
exp QuickSort O(nlgn)
HeapSort O(nlgn)
} we can't do better than this for comparison based sorts!

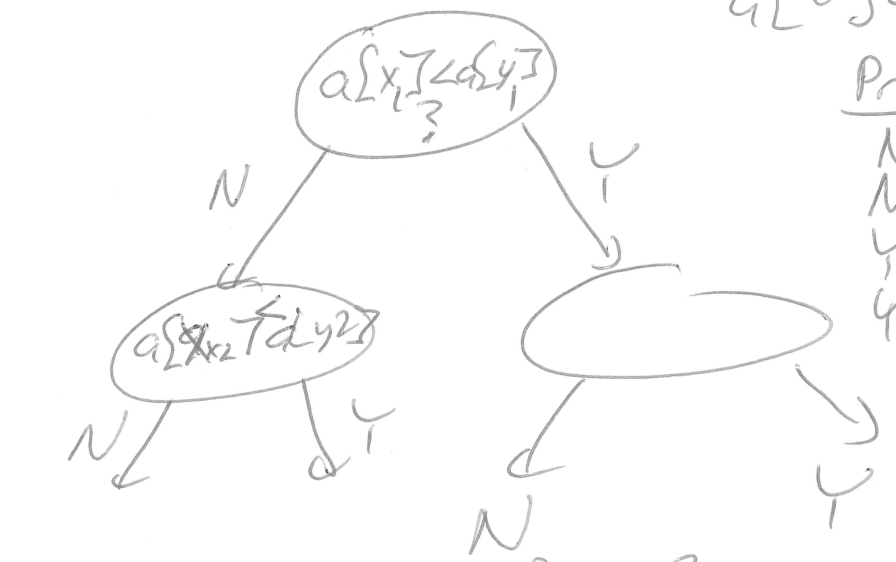
Input
1, 2, 3
1, 3, 2
2, 1, 3
2, 3, 1
3, 1, 2
3, 2, 1

Input can come in n! ways

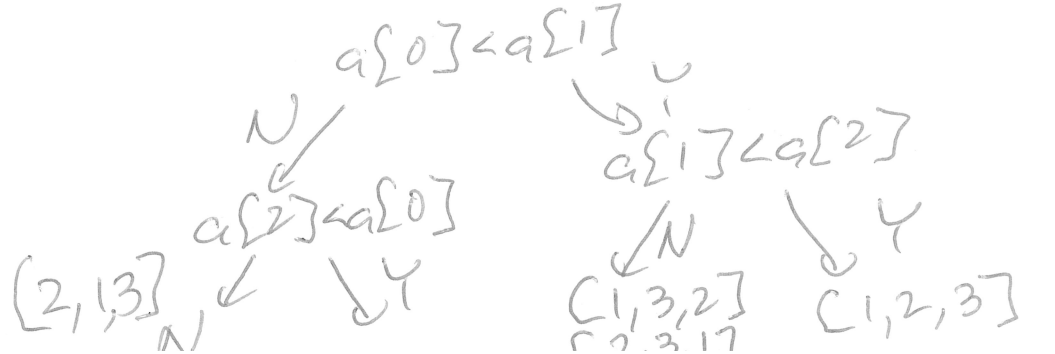
If an alg sorts all of these, it MUST make a different decision somewhere in its decision tree.

a[0] a[1] a[2]

Program
NN
NY
YN
YY



Limit prog to do 2 comparisons w/ 6 unique inputs, there will be 2 diff inputs that go down the same decision tree.



If we are to distinguish btw X different inputs, we must have at least X different computation paths.

Comparison Based Sorting (sorting n items)

inputs $= n!$

decision paths for k comparisons $= 2^k$

Stirling's Approx

$$2^k \geq n!$$

$$2^k \geq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

$$\log_2 2^k \geq \log_2 \sqrt{2\pi n} + \log_2 \left(\frac{n}{e}\right)^n$$

$$k \geq \underbrace{\text{small}}_{O(\lg n)} + n \log_2\left(\frac{n}{e}\right)$$

Const

$$k \geq O(\lg n) + n(\log_2 n - \log_2 e)$$

$$k \geq O(\lg n) + O(n \lg n)$$

What's the fewest # of comp necessary to sort 8 #s?

$8! = 40{,}320$

$2^k > 40{,}320$

$k > \log_2 40320$

$k = 16$

$2^{16} = 65536$

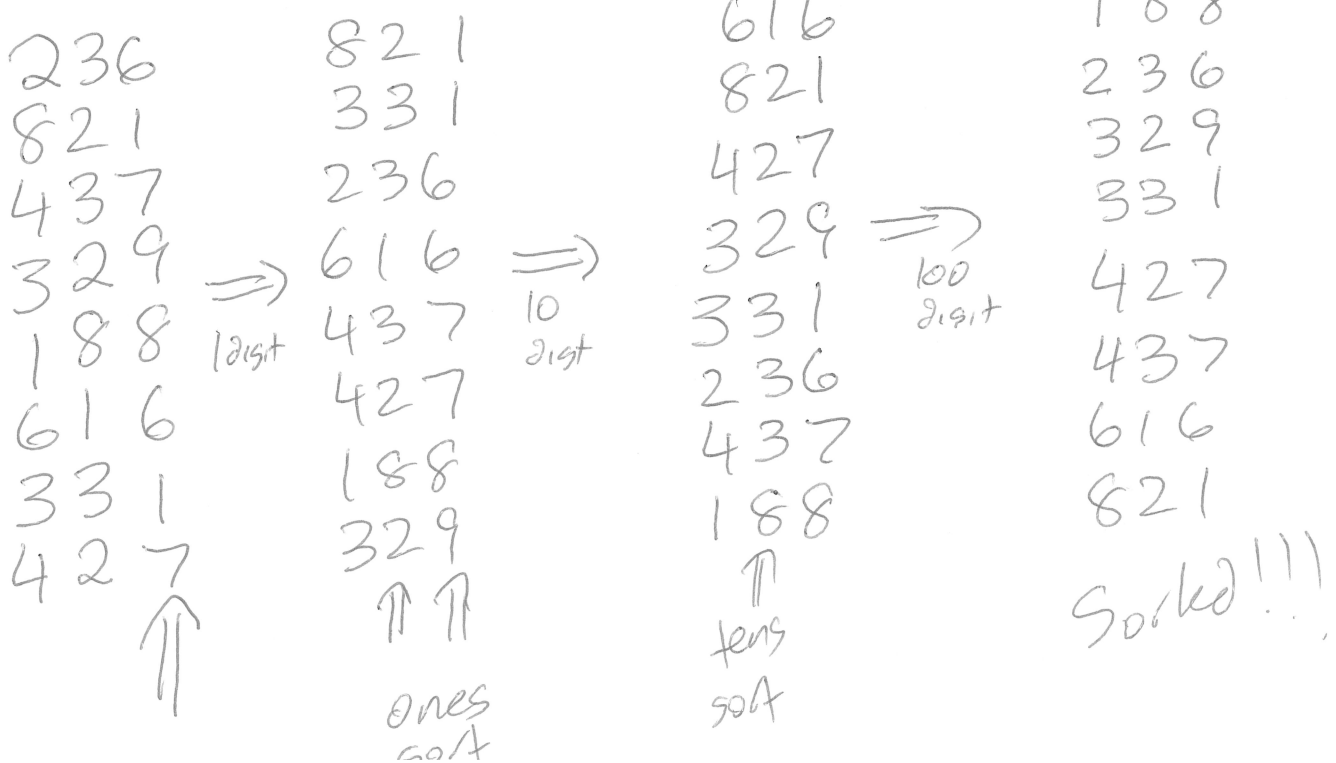
$2^{15} = 32768$

$15! \sim 3.2 \times 10^{12}$ (10)

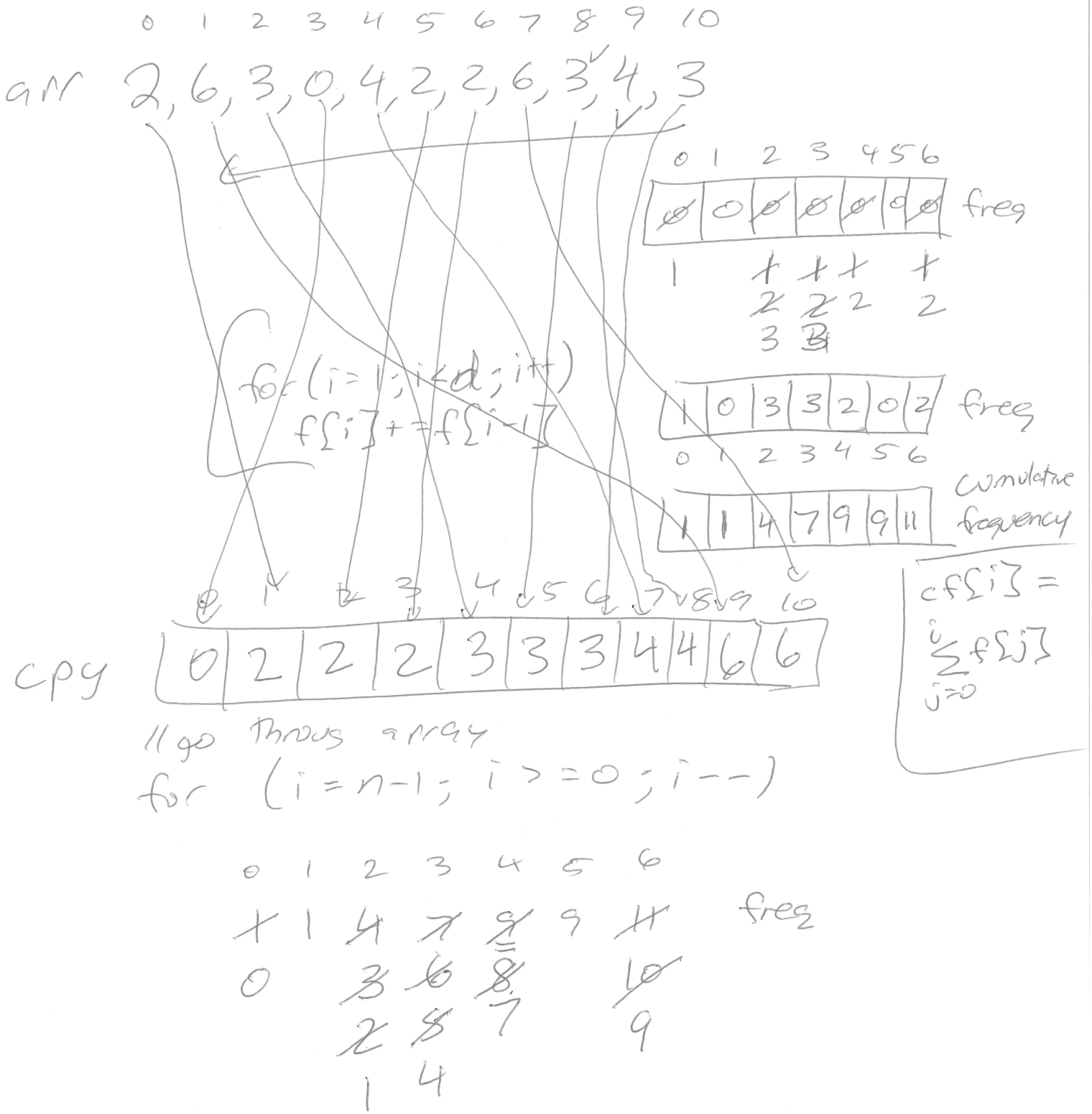
$2^{37} \approx 4 \times 10^{9}$

Radix Sort

Works on arrays of digits



Intermediate Sort to sort by digit is Counting Sort. Counting Sort is stable



Greedy Algorithms

In CS1, we did brute force, followed up w/backtracking for both it's easy to prove correctness, but the run-time quickly get unmanageable!

Some problems where we can PROVE that it's not necessary to consider all options to find the best one.

An algorithm that correctly solves a problem by only considering the "best option in the moment" as opposed to all options is called a greedy algorithm.

Planting Trees

4 → # trees

2 3 4 3 → how long to grow

4 3 3 2
1 2 3 4

5 5 6 6 ⟹ 7

Greedy choice
plant the trees that take longer 1st.

↔

2 (4)
1 3

exchange arg
if you're out of order I can always