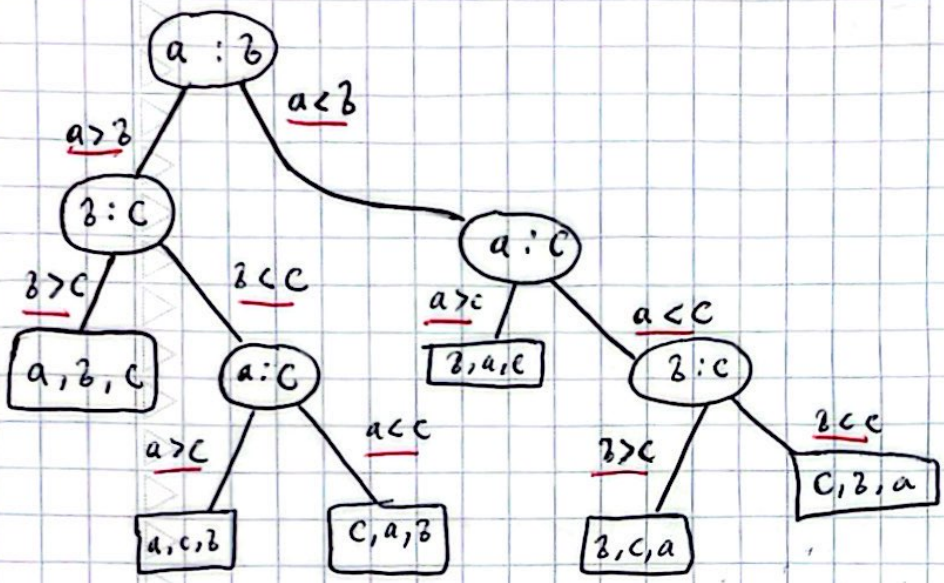
CS-21 LECTURE #5

Decision Tree for sorting a, b, c



minimum of $n!$ leaves

height $h = 2^n$ leaves

$h \geq \lg(n!)$

$h \geq (n \lg n)$

→ This is why comparison based sorts are lower bounded at $O(n \lg(n))$

Counting Sort

Seward - 1954

Input ↓ output ↓

```
Counting Sort (Array A, Array B, range k)
    new array C[k+1] = {0};
voting section [ for (i=0; i< A.length; i++) {
                     C[A[i]]++;
sum the values to find indexes [ for (i=1; i<=k; i++)
                                     C[i] += C[i-1];
```

- Assume each of n elements is in the range $0 \to k$. As long as k is about $O(n)$ then $O(n)$ sort

Best case:

$k < n$ $k << n$

```
for (i = A.length -1; i >= 0 ; i --) {
    B[C[A[i]] -1] = A[i];
    C[A[i]]--;
}
}
```

The sort (based on accumulated knowledge)

Stable Sort

- A sort that preserves order in case of ties. Counting sort does ~~that~~ this

Stable Sorts

- Bubble Sort.
- Insertion Sort.
- Merge Sort.
- Counting Sort.
- Bucket Sort.

(UN)(NON) Stable

- selection sort.
- quick sort.
- Heap Sort.

Radix Sort - 1 'digit' at a time from LSB to MSB

Combine
1929

Array

s - # of digits

k - possible range for each digit.

```
Radix (A, s) {
    for (i=0; i<s; i++) {
        Use an "O(n)" stable sort to sort array A on digit i
```

Bucket Sort

Isaact Singleton - 1956

```
BucketSort (A, n)
    for (i=0; i<n; i++) {
        insert A[i] into bucket B[A[i]]
    for (i=0; i<n; i++) {
        sort bucket B[i] with insertion sort
    combine B[0], B[1], ..., B[n-1] back into array
```

Radix Bucket sort = Divide into buckets, combine, on and on and on until sorted.

Radix - Bucket Sort

* IMPORTANT *

Radix Bucket Sort (A, n, s, k) {

Array size

Not parameters?

s = # of "digits"

k = radix of "digits"

~~n = A.length()~~

new Buckets[k]; // k is 10 for range of digits

Radix part

for (i=1; i<=s; i++) { // ith digit

for (j=0; j<n; j++) {

r = ith digit of A[j]

bucket[r].insertBack(A[j]);

} for j

for (j=0; j<k; j++) {

add items from Bucket[j] into A

clear Bucket[j];

} for j

} for i

} end function

EX: SSN ints

digits s = 9

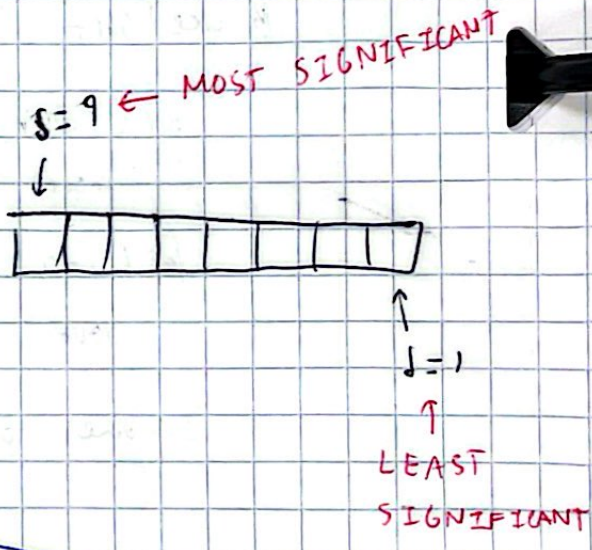
radix k = 10 (digits are 0-9)

$O(s(n+k))$

Best when

$s << n$

$k << n$



DONT USE LINKED LIST FOR ASSIGNMENT USE VECTORS