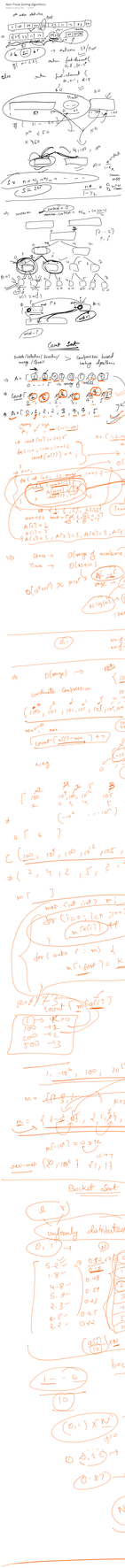
Non-Trivial Sorting Algorithms
Nishant Kumar Singh



Count Sort

Bucket/Pidhon/Bucket/ counting/Count > Customizes based sorting algorithms

$A = [\ ]$

$A[i] \to \frac{i}{5}, \frac{i}{5}, \frac{i}{5}, \frac{i}{5}, \frac{i}{5}$

count / freq / $\frac{A}{k}$ = (n-1-A[i])

id count[A[i]] = 0

for (int i = 1; i < n; i++) { count[A[i]] ++ } → O(n)

for (int i = 1; i < n; i++) { int idx = n-1-count[A[i]]; ans[idx] = A[i]; count[A[i]]--; } → O(n)

O(n)

A[i] = 5, 5, 5, 5, 3, 3, 3, 3, 5

count[3] = 4, count[5] = 5

A[0] = 5

A[1] = 5

A[2] = 5

A[3] = 5, A[4] = 3, A[5] = 3, A[6] = 3, A[7] = 5

time → O(range of numbers) ≈ O(n)

Time → O(n+k)

O(n+k) ≈ n+k

Radix Sort

A = [901, 423, 42, 1, 87]

→ 901, 42, 423, 1, 87

→ 901, 01, 423, 01, 87

→ 1, 24, 87, 423, 901