Lecture 10: Sorting III: Linear Bounds
Linear-Time Sorting

Lecture Overview

- Sorting lower bounds
  - Decision Trees
- Linear-Time Sorting
  - Counting Sort

Readings

CLRS 8.1-8.4

Comparison Sorting

Insertion sort, merge sort and heap sort are all comparison sorts. The best worst case running time we know is \( O(n \lg n) \). Can we do better?

Decision-Tree Example

Sort \(<a_1, a_2, \cdots, a_n>\).

Each internal node labeled \(i : j\), compare \(a_i\) and \(a_j\), go left if \(a_i \leq a_j\), go right otherwise.
Example

Sort \(< a_1, a_2, a_3 > = < 9, 4, 6 >\) Each leaf contains a permutation, i.e., a total ordering.

![Decision Tree Execution](image)

**Decision Tree Model**

Can model execution of any comparison sort. In order to sort, we need to generate a total ordering of elements.

- One tree size for each input size \(n\)
- Running time of algo: length of path taken
- Worst-case running time: height of the tree

**Theorem**

Any decision tree that can sort \(n\) elements must have height \(\Omega(n \lg n)\).

**Proof:** Tree must contain \(\geq n!\) leaves since there are \(n!\) possible permutations. A height-\(h\) binary tree has \(\leq 2^h\) leaves. Thus,

\[
\begin{align*}
n! & \leq 2^h \\
\implies h & \geq \log(n!) \quad (\geq \log((n/e)^n) \text{ Stirling}) \\
& \geq n \log n - n \log e \\
& = \Omega(n \log n)
\end{align*}
\]
Sorting in Linear Time

Counting Sort: no comparisons between elements

Input: $A[1 \ldots n]$ where $A[j] \in \{1, 2, \ldots, k\}$

Output: $B[1 \ldots n]$ sorted

Auxiliary Storage: $C[1 \ldots k]$

Intuition

Since elements are in the range $\{1, 2, \ldots, k\}$, imagine collecting all the $j$’s such that $A[j] = 1$, then the $j$’s such that $A[j] = 2$, etc.

Don’t compare elements, so it is not a comparison sort!

$A[j]$’s index into appropriate positions.

Pseudo Code and Analysis

\[
\begin{align*}
\text{for } i & \leftarrow 1 \text{ to } k \\
& \text{do } C[i] = 0 \\
\text{for } j & \leftarrow 1 \text{ to } n \\
& \text{do } C[A[j]] = C[A[j]] + 1 \\
\text{for } i & \leftarrow 2 \text{ to } k \\
& \text{do } C[i] = C[i] + C[i-1] \\
\text{for } j & \leftarrow n \text{ downto } 1 \\
& \text{do } B[C[A[j]]] = A[j] \\
& \quad C[A[j]] = C[A[j]] - 1
\end{align*}
\]

$\theta(n+k)$
Example

Note: Records may be associated with the $A[i]$’s.

\begin{figure}[h]
\centering
\begin{tabular}{cccc}
A: & 1 & 2 & 3 & 4 & 5 \\
& 4 & 1 & 3 & 4 & 3 \\
& 1 & 2 & 3 & 4 & 5 \\
B: & 1 & 3 & 3 & 4 & 4 \\
& 1 & 2 & 3 & 4 \\
C: & 0 & 0 & 0 & 0 \\
& 1 & 0 & 2 & 2 \\
& 1 & 1 & 2 & 4 \\
\end{tabular}
\end{figure}

Figure 4: Counting Sort Execution

\begin{align*}
C[3] &= 3 \\
C[4] &= 5 \\
\end{align*}