Week 15 Summary: Disjoint Set Union and Amortized Analysis

Z5007 Programming and Data Structures · MTech DSAI · Odd Semester 2026-27 · Dr. Innocent Nyalala

1. Three Methods of Amortized Analysis

- **Aggregate:** Bound total cost $T(n)$ across n operations and divide by n (e.g., n dynamic array appends copy $< 2n$ items, giving $O(1)$ amortized cost).
- **Accounting:** Assign amortized charges (credits) stored on elements to pay for later expensive operations (setting bit $0 \to 1$ pays 2 credits: 1 to flip now, 1 saved for its future reset $1 \to 0$).
- **Potential Function:** Define $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$ with $\Phi(D_n) \geq \Phi(D_0)$ so $\sum \hat{c}_i \geq \sum c_i$.

2. Union-Find with Path Compression and Union by Size

Attaching the smaller tree under the larger tree keeps tree depth $\leq \lfloor \log_2 n \rfloor$. Adding path compression inside `find(x)` flattens traversed nodes straight to the root, reducing amortized cost per operation to $O(\alpha(n))$.

Worked Check

Across $1,024$ increments of a binary counter, actual bit flips equal $2,037 < 2 \times 1024$ with $\Phi = 1$ remaining one-bit.