Recap of this week’s lectures:

- Amortization (Binary Counter, Dictionary)
- Splay Trees

**Binary Counter Revisited:** Suppose we are incrementing a binary counter, but instead of each bit flip costing 1, suppose flipping the $i^{th}$ bit costs us $2^i$. (Flipping the lowest order bit $A[0]$ costs $2^0 = 1$, the next higher order bit $A[1]$ costs $2^1 = 2$, the next costs $2^2 = 4$, etc.) What is the amortized cost per operation for a sequence of $n$ increments, starting from zero?

**Solution:** $O(\log n)$. The idea is simple. We flip $A[0]$ each time, so pay $n$ over $n$ operations. We flip $A[1]$ every other time, so pay $\leq 2 \times n/2 = n$ over $n$ operations, and so on, until $A[\lceil \log_2 n \rceil]$ which gets flipped once for a cost of at most $n$. Hence $O(n \log n)$ in total, or $O(\log n)$ per operation.
**Spray paint:** At the FTW Motor Company, there’s an infinite line of cars, each of which are initially colored white. Associate the cars with the integers, both positive and negative. Management sends the paint crew a sequence of Spray commands: each command is of the form \texttt{Spray}(x, y, c) (where \(x \leq y\)), which requires spray-painting all the cars/integers in the interval \([x, y]\) with the color \(c\). The cost of each operation \texttt{Spray}(x, y, c) is the number of distinct colors in the range \([x, y]\) before the operation is performed.

Show using a potential function that the cost of \(N\) paint operations is at most \(3N\).

**Solution:** Define \(\Phi\) as the number of adjacent cars with different color.

\(\Phi\) starts out at 0, and clearly cannot ever be smaller than 0.

We need to show that for every operation \texttt{Spray}(x, y, c), \(C + \Delta\Phi \leq 3\), where \(C\) is the cost of the operation: the number of distinct colors in the range \([x, y]\).

Note that the number of pairs of adjacent cars in the range \([x, y]\) that are of different color is at least \(C - 1\), because there are \(C\) distinct colors in the range \([x, y]\). After the operations, all cars in the range \([x, y]\) are the same color, causing \(\Phi\) to decrease by at least \(C - 1\), so just from this, \(\Delta\Phi \leq -C + 1\). However the operation could cause the relation between \(x - 1\) and \(x\), and \(y\) and \(y + 1\), to change. This changes \(\Delta\Phi\) by some integer in the range \([-2, 2]\), so \(\Delta\Phi \leq -C + 1 + 2 = -C + 3\).

Thus \(C + \Delta\Phi \leq C - C + 3 = 3\) as desired.

Finally, let \(\Phi_0\) and \(\Phi_N\) be the initial and final values of \(\Phi\), respectively. Clearly \(\Phi_N - \Phi_0 \geq 0\), because we started out with the minimum possible value of \(\Phi\). So summing this over all operations gives \(\text{totalcost} + \Phi_N - \Phi_0 \leq 3N \implies \text{totalcost} \leq 3N\).
Cyclic Splaying: Starting from a tree $T_0$ of $n$ nodes a sequence of $\ell \geq 1$ splay operations is done. It turns out that the initial tree $T_0$ and the final tree $T_\ell$ are the same. Let $k$ be the number of distinct nodes splayed in this sequence. (Clearly $k \leq \ell$.) Below is an example where $k = \ell = 4$ and $n = 6$.

4 3 0 5 4
/ \ / \ / /
2 5 splay(3) 2 4 splay(0) 2 splay(5) 2 splay(4) 2 5
/ \ =========> / \ =========> / \ =========> / \
0 3 0 5 1 3 / \
\ \ \
1 1 4 0 4 1
\ / / \\
5 1 2

(a) Use some setting of node weights to show that the average number of splaying steps in this cycle (i.e., the average per splay operation) is at most $1 + 3 \log_2 n$. Make use of the Access Lemma for splay trees covered in lecture yesterday.

(b) (Extra material, for you to do at home.) Now use a different setting of node weights to show that the average number of splaying steps in this cycle (per splay operation) is at also most $1 + 3 \log_2 k$.

Solution: Start with equation (2) of the lecture on amortized analysis:

$$\sum_i c_i = \left( \sum_i ac_i \right) + \Phi(s_{\text{initial}}) - \Phi(s_{\text{final}})$$

Note that since the initial and final trees are the same, the initial and final potentials are equal. So the above equation shows that the total cost is the same as the total amortized cost.

By the Access Lemma, the amortized number of splaying steps in each splay is $3(r(t) - r(x)) + 1 \leq 3r(t) + 1$. (Since all the weights are positive.) So the total amortized cost, which equals the total actual number of splay steps is at most $\ell(3r(t) + 1)$, and the average number of splay steps per splay is at most $(3r(t) + 1)$.

a. Set all the weights of each node to 1. We know $r(t)$, or the rank of the root, is $\lfloor \log(s(t)) \rfloor$, where $s(t) = \sum_{y \in T(t)} w(y)$. Since all $n$ nodes are in the subtree rooted at the root, $s(t) = n$ and $r(t) = \lfloor \log(n) \rfloor$. We also know $r(x) \geq 0$, so $3(r(t) - r(x)) + 1 \leq 3\lfloor \log(n) \rfloor + 1$.

b. Let the weights of the $k$ elements we access be 1 and the others be $\epsilon > 0$. The rank of the root is then $\lfloor \log(k + (n - k)\epsilon) \rfloor$. If we choose $\epsilon$ to be small enough so that $(n - k)\epsilon$ is smaller than 1, then $\lfloor \log(k + (n - k)\epsilon) \rfloor = \lfloor \log(k) \rfloor \leq \log(k)$. This is because the function $f(x) = \lfloor \log(x) \rfloor$ only changes when $x$ crosses an integer value. Again by the Access Lemma, we know the average number of splaying steps is $\leq 3(\log(k) - r(x)) + 1 \leq 3\log(k) + 1$. 

3
**Balls in Bins** There are $n$ balls and an infinite number of bins. A bin can have 0 or more balls in it. A move consists taking all the balls of some bin and putting them into distinct bins. The cost of a move is the number of balls moved. Define the potential of a state of this system as the sum of the potentials of all the bins. The potential of a bin with $k$ balls in it is:

$$
\Phi(k) = \max(0, k - z)
$$

Where for convenience $z = \lfloor \sqrt{n} \rfloor$.

1. Prove that the amortized cost of a move is at most $2z$.

**Solution:** Consider one move in which $k$ is the number of balls in the bin vacated. Let $ac$ be the amortized cost of the move. Let $X$ be the set of bins whose count increases where the final number of balls is $>z$. (These are precisely the bins causing the potential to increase.) Breaking the analysis into two cases we get:

$$
ac = \begin{cases} 
  k + |X| & \text{if } k \leq z \\
  k - (k - z) + |X| & \text{otherwise}
\end{cases}
$$

Clearly $|X| \leq k$, so in the first case $ac \leq 2k \leq 2z$. We also know that each bin of $X$ has at least $z + 1$ balls in it at the end. So we know that $|X| * (z + 1) \leq n$. So $|X| \leq z$. (If $|X| \geq z + 1$ then $|X| * (z + 1) > n$, a contradiction.) So in the 2nd case we have:

$$
ac = k - (k - z) + |X| \leq k - (k - z) + z = 2z
$$

which finishes the proof. ■

2. Show a sequence of moves which achieves $\Omega(z)$ per move.

**Solution:** Let $k = \lfloor \sqrt{2n} \rfloor$. With this choice $k + (k - 1) + \ldots + 1 \leq n$. Move balls around until you’ve created a bin with $k$ balls, one with $k - 1$ balls, etc., down to a bin with 1 ball. Now take the bin with $k$ balls in it, and move those balls to a bin with $k - 1$ balls, $k - 2$ balls, etc, plus one ball to an empty bin. The new arrangement is isomorphic to the previous one. So this move, which costs $k$, can be repeated forever.