Pancake flipping: optimal stack-sorting algorithms

Aaron Cho, Kailey Hua
Advisor: Dr. Susan Durst

Proof School Math Burst 2019

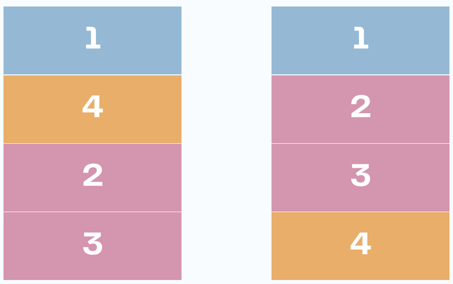
Problem statement:

- Given a disordered stack of n pancakes, each of different size (assume integer sizes without loss of generality), sort the stack from smallest to largest with the smallest pancake on top and the largest pancake on the bottom
- Pancakes may be flipped by inserting the spatula between two pancakes in the stack and "flipping" the stack above the spatula

01

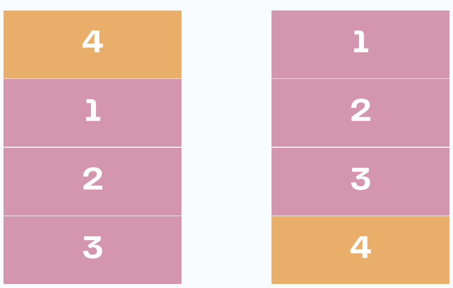
Simple algorithm

- Find the largest pancake that is not in its correct spot in its desired stack



current stack desired stack

- Find the largest pancake that is not in its correct spot in its desired stack (orange)
- Cut the stack and flip it to the top



- Flip it to the desired place (ex. if it should be the third pancake from the top, flip between pancakes 3 and 4)
- Repeat

Analysis of the simple algorithm

The worst possible stack of n pancakes takes 2n-3 flips to solve.

Corollary: If every stack of n-1 pancakes can be solved in at most f flips, then the upper bound of flips for any stack of n pancakes is f+2.

Corollary: A stack with no consecutive connected pancakes in which the largest pancake is not on the bottom can be solved in at least n steps.

02

Generating the worst possible stack

There are n-2 possible worst stacks for the algorithm. They are generated via the following pattern:

- For **odd n** stack the pancakes 1, n, n-2, n-4, n-6, ..., 3, 4, ..., n-7, n-5, n-3, n-1, 2
- For **even n** stack the pancakes 2, n, n-2, n-4, n-6, ..., 4, 3, ..., n-7, n-5, n-3, n-1, 1

Order of pancakes to flip to solve worst stacks:

- For odd n: n-1, n-2, n-3, ..., 3, n, 2, 1
- For even n: n-1, n-2, n-3, ..., 3, n, 1

03

Proposed optimal algorithm

The basic idea of our proposed algorithm is to put pancakes next to pancakes they should be next to.

1. If biggest pancake is on top, flip entire stack
2. Put the top pancake next to a consecutive pancake without breaking consecutive stacks
    a. If there is more than one option, create a bigger consecutive stack
    b. If both options are the same, flip onto the one closer to bottom
    c. Otherwise, choose the larger number
3. If there is no way to perform step 2 optimally, revert to the simple algorithm
4. If this stack is still on top, flip the entire stack
5. Repeat

04

Advantages of the proposed algorithm

- It is rare but possible for our proposed algorithm to be stuck in an infinite loop
- Always does the "intuitive" thing, while the simple algorithm performs counterintuitive and redundant operations
- Faster than the simple algorithm
- Considers and preserves consecutive stacks of pancakes

05

Proof of the optimality of the proposed algorithm

- After each flip, at most one pair of consecutive numbers that were not next to each other can now be next to each other
- This means that since there are n-1 connections between pancakes, the lower bound for the number of flips to solve is n-1
- However, since one pair of consecutive numbers (for example, 3 and 4) are already next to each other, we know we cannot solve these stacks in any less then n-2 flips
- For even numbers, we know our algorithm solves the stack in n-1 flips
- This is one more than the supposed minimum, and the reason for this is that we need to take in an additional requirement
- This requirement is that 1 must be on the top
    - 1 starts on the bottom for least optimal even stacks, and the only way to change this is by flipping the entire stack
    - However, flipping the entire stack does not change which pancakes are next to each other, so the minimum number of flips for even numbers is actually (n-2) + 1 = n-1.
- For odd numbers the lower bound is n-2 but our algorithm solves the stack in n flips.
    - 2 starts on the bottom, and we must flip it to the top
- After we flip 2 to the top, we need to flip the top two pancakes to bring 2 to the second position
- Now, the only way to reach this lower bound is if every flip connects two consecutive numbers, and if after 2 is flipped to the top, the stack is one move from solved
- The only way to reach the second condition is if we can arrange the stack in the order n, n-1, n-2, ..., 3, 1, 2, in n-3 moves
- The only way to do this is to connect two consecutive numbers with every flip
    - So our first step would be forced
- The original stack is 1, n, n-2, n-4, n-6, ..., 3, 4, ..., n-7, n-5, n-3, n-1, 2, and the only move that connects two consecutive numbers is flipping the 1 onto the 2, leaving the stack n-1, n-3, n-5, ..., 4, 3, ..., n-6, n-4, n-2, n, 1, 2
- However, this causes a contradiction since we need to bring the n to the top
- The only way to do this is by flipping the top pancake on top of the 1, but the only pancake that is consecutive to the 1 is the 2, which is beneath the 1, and cannot be brought above without flipping the entire stack which we know cannot connect two consecutive numbers
- So therefore there must be an additional flip, changing our lower bound from n-1 to n, which is the number of moves that our algorithm solves the stack in.