CS2040S Recitation 2

AY 24/25 Sem 2 — github/omgeta

Q1. (a.) Algorithm for $2(n-2)+1 = 2n-3$ flips:

1. Consider all panckages skipping i from the bottom, where i starts off as 0
2. If there is only 1 pancake left, we are done.
3. Find the largest pancake in the substack
4. If the pancake is position j, where j is not the bottom of the substack, flip pancakes $[j, n]$ and then $[i+1, n]$ so the largest pancake is now at the bottom of the substack
5. Increment i by 1, and repeat from step 1

(b.) Loop invariant is at the end of every ith iteration, the largest i pancakes are in their correct position. This is most analogous to SelectionSort.

(c.) $3(n-2)+2 = 3n-4$

(d.) $2N$, where N is min(number of large pancakes, number of small pancakges)

(e.) Same as in (a), $2n-3$

(f.) Consider the case where the largest panckage is placed below the stack, the second largest at the top, and the process continuing to alternate between decreasing and increasing sizes. We need atleast n flips in this case. This is called a lower bound.