MaximizePayout

Benjamin Katz

March 2022

1 Code

I simply sorted both lists and matched the smallest base with the smallest exponent and multiplied it to the total and continue through the list. This is a greedy algorithm because it makes a local decision at every step, that local decision being - match with the next smallest base with the next smallest exponent. Intuitively, this maximizes payout because we multiply together a lot of large bases and few small bases.

2 Proof

Because of the commutative property of multiplication we can present the greedy algorithm as first matching the largest base with the largest exponent and make its way down to the smallest base and the smallest exponent.

Proof by Induction

Let $f(a_i, b_i) = \prod a_i^{b_i}$ in the greedy algorithm

Let $f^*(a_i, b_i) = \prod a_i^{b_i}$ in the optimal algorithm

- Base Case: $f(a_1, b_1) >= f^*(a_x, b_y)$
- By inspection, the greatest single term must be the largest base raised to the largest exponent. Therefore the first term of greedy must be greater than or equal to any single term in optimal
- Because of the commutative property, the optimal could be arranged in any order so instead of using subscripts x and y we can replace it with i and i and assume that the ordering of the lists reflects the optimality of f^*
- Inductive Hypothesis: assume that $f(a_i, b_i) >= f^*(a_i, b_i)$
- Prove: $f(a_{i+1}, b_{i+1}) >= f^*(a_{i+1}, b_{i+1})$ for all values of i
- (Equation one):$f(a_{i+1}, b_{i+1}) = f(a_i, b_i) * a_{i+1}^{b_{i+1}}$ for greedy

- Because the greedy algorithm works in descending order, $a_{i+1}^{b_{i+1}}$ is the lowest term in the greedy algorithm and is the $(i+1)^{th}$ highest possible base-exponent combination
- (Lemma)Therefore the lowest term in optimal $f^*(a_{i+1}, b_{i+1})$ must be less than or equal to the greedy $a_{i+1}^{b_{i+1}}$
- Because of the commutative property we can write the optimal by reordering and factoring out the lowest term (Equation two):$f^*(a_{i+1}, b_{i+1}) = f(a_i, b_i) * lowestTerm$
- Since the $f(a_i, b_i) >= f^*(a_i, b_i)$ by the inductive hypothesis
- And as the lemma states $a_{i+1}^{b_{i+1}} >= lowestTerm$
- Therefore $f(a_i, b_i) * a_{i+1}^{b_{i+1}} >= f(a_i, b_i) * lowestTerm$
- Looking at equations one and two we can rewrite the above equation as $f(a_{i+1}, b_{i+1}) >= f^*(a_{i+1}, b_{i+1})$ completing the induction.
- The greedy solution is optimal QED

3 Time Complexity

- Sorting both lists is an $O(n \log n)$ operation (See slides on sorting)
- Iterating through the two list $O(n)$ time doing a constant amount of work at every point in the list(a power and multiply)
- $2n \log n + O(n) = O(n \log n)$