CS2040S Recitation 1

AY 24/25 Sem 2 — github/omgeta

Q1. (a.) `isDivisible(n, i)` is $O(1) \implies$ `isPrime(n)` is $O(n)$ ■
`isDivisible(n, i)` is $O(n) \implies$ `isPrime(n)` is $O(n^2)$ ■
`isDivisible(n, i)` is $O(i) \implies$ `isPrime(n)` is $O(n^2)$ ■

(b.) `isDivisible(n, i)` is $O(n) \implies$ `isPrime2(n)` is $O(n\sqrt{n})$ ■
`isDivisible(n, i)` is $O(i) \implies$ `isPrime2(n)` is $O(n)$ ■

(c.) $T(n) = \sum_{i=2}^{\sqrt{n}} (T(i) + O(1))$ ■

Q2. (a.)
1. Let the sum of salaries be K, where K is a large random value
2. Each student i adds their salary s_i to the sum before passing current sum to the next student
3. At the end, student 1 removes the random number K and divides the sum by n to find the average salary

This is secure because each student only knows the sum of the previous $n-1$ studnets with the random number without any specific information revealed. ■

(b.) The algorithm is vulnerable to the first student having info on the secret value.

1. Let the sum of salaries start at 0
2. Each student i adds $s_i + k_i$, where s_i is their salary and k_i a private secret value to the current sum before passing to the next student
3. After one round, each student removes their secret value from the sum
4. After the second round, everyone will know the total sum of salaries which divided by n gives the average salary

This fix solves the issue giving the first student advantageous information if colluding with other students. However, there is a limitation if $n-1$ students all collude to share their group total sum, then the remaining student will always have his salary known. ■

(c.) During the second round, the saboteur could add/subtract a value which was not his initial seed. ■