Names and emails of everyone in Project Group 9:
Xander Lahti (xanderla@csu.fullerton.edu)
Xareni Merino Rita (xmerino@csu.fullerton.edu)
Te Yen Lee (leeteyen91@csu.fullerton.edu)
Juan Martinez Vasquez (diegomarvas@csu.fullerton.edu)

PSEUDOCODE FOR ALGORITHM 1:

```
function minSwaps(row){
        n = length(row) / 2
        position = new vector array (size 2n)

        for i = 0 to 2n{
                position[row[i]] = i
        }

        swaps = 0

        for i = 0 to 2n(by increments of 2){
                currentPerson = row[i]
                partner = currentPerson XOR 1

                if row[i + 1] != partner{
                        swaps += 1

                        partnerPosition = position[partner]

                        swap(row[i + 1], position[partner])

                        position[row[partnerPosition]] = partnerPosition
                        position[partner] = i + 1
                }
        }

        return swaps
}
```

PROVING EFFICIENCY OF ALGORITHM 1 PSEUDOCODE

```
function minSwaps(row){
	// constant time, so thus O(1)
	n = length(row) / 2

	// time proportional to size of array, which is 2n, so thus O(n)
	position = new vector array (size 2n)

	// since it runs to 2n, it is O(n)
	for i = 0 to 2n{
		position[row[i]] = i
	}

	// constant time, so thus O(1)
	swaps = 0

	// since it runs to 2n and all within is constant time of O(1), this loop is O(8n)
	for i = 0 to 2n(by increments of 2){

		// constant time, O(1)
		currentPerson = row[i]

		// constant time, O(1)
		partner = currentPerson XOR 1

		// constant time, O(1)
		if row[i + 1] != partner{

			// constant time, O(1)
			swaps += 1

			// constant time, O(1)
			partnerPosition = position[partner]

				// constant time, O(1)
			swap(row[i + 1], position[partner])

			// constant time, O(1)
			position[row[partnerPosition]] = partnerPosition

			// constant time, O(1)
```

```
                position[partner] = i + 1
            }
        }

        // constant time, O(1)
        return swaps
}
```

Thus it seems that the pseudocode above for Algorithm 1 is O(8n) + O(n) + O(n) + O(3) which equals to O(10n + 3) = O(n)

PROOF BY LIMITS

$T(n) = 10n + 3 \in O(n)$,

$\lim_{n \to \infty}(T(n) / f(n))$
$= \lim_{n \to \infty}((10n + 3) / (n))$
$= \lim_{n \to \infty}((10n / n) + (3 / n))$
$= 10 + 0 = 10$

Thus, since the limit is constant, it shows that $T(n) \in O(n)$, and thus by proof of limits, O(n) is the time complexity of the pseudocode written above.