Chapter 5

Reserves Recitation.

5.1

Another sorting algorithms, that it's correctness isn't so obivoius.

Result: returns the multiplication $x \cdot y$ where $x, y \in \mathbb{F}_2^n$

1 **for** $i \in [n]$ **do**
2   **for** $j \in [n]$ **do**
3   **if** $A_j < A_i$ **then**
4   swap $A_i \leftrightarrow A_j$
5   **end**
6   **end**
7 **end**

Claim 5.1.1. *After the ith iteration, $A_1 \leq A_2 \leq A_3.. \leq A_i$ and A_i is the maximum of the whole array.*

Proof. By induction on the iteration number i.

1. Base. For $i = 1$, it is clear that when j reaches the position of the maximal element, an exchange will occur and A_1 will be set to be the maximal element. Thus, the condition on line (3) will not be satisfied until the end of the inner loop and indeed, we have that A_1 at the end of the first iteration is the maximum.

2. Assumption. Assume the correctness of the claim for any $i' < i$.

3. Step. Consider the ith iteration. And observe that if $A_i = A_{i-1}$ then A_i is also the maximal element in A, namely no exchange will be made in the ith iteration, yet $A_1 \leq A_2 \leq .. \leq A_{i-1}$ by the induction assumption, thus $A_1 \leq A_2 \leq .. \leq A_{i-1} \leq A_i$ and A_i is the maximal element, so the claim holds in the end of the iteration. If $A_i < A_{i-1}$ then there exists $k \in [1, i-1]$ such that $A_k > A_i$. Set k to be the minimal position for which the inequality holds. For convenience, denote by $A^{(j)}$ the array in the beginning of the jth iteration of the inner loop. And let's split into cases according to j value.

(a) $j < k$ By definition of k, for any $j < k$, $A_j^{(1)} < A_i^{(1)}$, Hence in the first $k-1$ iterations no exchange will be made and we can conclude that $A_l^{(j)} = A_l^{(1)}$ for any $l \in [n]$ and $j \leq k$.

(b) $j \geq k$ and $j < i+1$, We claim that for each such j an exchange will always occur.

Claim 5.1.2. *For any $j \in [k,i]$ we have that in the end of the jth iteration:*

- $A_j^{(j+1)} = A_i^{(j)}$.
- $A_i^{(j+1)} = A_j^{(j)} = A_j^{(1)}$.
- *For any $l > j$ and $l \neq i$ we have $A_l^{(j+1)} = A_l^{(1)}$.*

(c) $j > i$, so we know that $A_i^{(i+1)}$ is the maximal element in A. Therefore, for any j, it holds that $A_i^{(i+1)} \geq A_j^{(i)}$. It follows that no exchange would be made and $A_i^{(j)}$ will remain the maximum until the end of the inner loop.

$\square$

Of Claim 5.1.2. Observe that the third section holds trivially by the definition of the algorithm. It doesn't touch any position greater than j in the first j iterations (inner loop) except the ith position. So we have to prove only the first two bullets. Again, we are going to prove them by induction.

1. Base. $A_k^{(1)}$ is greater than A_i, and by the above case, we have that at the beginning of the kth iteration $A_k^{(k)} = A_k^{(1)}, A_i^{(k)} = A_k^{(1)}$. Therefore, the condition on line (3) is satisfied, an exchange is made, and $A_k^{(k+1)} = A_i^{(k)} = A_i^{(0)}$ and $A_i^{(k+1)} = A_k^{(k)}$. Now, $A_{k+1}^{(k+1)} = A_{k+1}^{(k)} = A_{k+1}^{(0)}$.

2. Assumption. Assume the correctness of the claim for any $k \geq j' < j \leq i$.

3. Step. Consider the $j \in (k,i]$ iteration. By the induction assumption, we have that $A_{j-1}^{(j)} = A_i^{(j-1)}$ and $A_i^{(j)} = A_{j-1}^{(j-1)} = A_{j-1}^{(1)}$. On the other hand, by the induction assumption of Claim 5.1.1, $j-1 < i \Rightarrow A_{j-1}^{(1)} \leq A_j^{(1)}$. Combining the third bullet, we obtain that:

$$A_j^{(j)} = A_j^{(1)} \geq A_{j-1}^{(1)} = A_i^{(j)}$$

And therefore, either there is an inequality and an exchange is made or there is equality. In both cases, after the ith iteration, we have $A_j^{(j+1)} = A_i^{(j)}$ and $A_i^{(j+1)} = A_j^{(j)} = A_j^{(1)}$.

$\square$

Result: returns the multiplication $x \cdot y$ where $x, y \in \mathbb{F}_2^n$

if $x, y \in \mathbb{F}_2$ **then**
    return $x \cdot y$
end

else
    define $x_l, x_r \leftarrow x$ and $y_l, y_r \leftarrow x$ // $O(n)$.

    calculate $z_0 \leftarrow \text{Karatsuba}\,(x_l, y_l)$
        $z_2 \leftarrow \text{Karatsuba}\,(x_r, y_r)$
        $z_1 \leftarrow \text{Karatsuba}\,(x_r + x_l, y_l + y_r) - z_0 - z_2$

    return $z_0 + 2^{\frac{n}{2}} z_1 + 2^n z_2$ // $O(n)$.
end