Q3

Assume you are given a map of a straight sea shore of length 100n meters as a sequence on 100n numbers such that Ai is the number of fish between i th meter of the shore and (i + 1)th meter, 0 ≤ i ≤ 100n − 1. You also have a net of length m meters but unfortunately it has holes in it. Such a net is described as a sequence N of n ones and zeros, where 0's denote where the holes are. If you throw such a net starting at meter k and ending at meter k + n, then you will catch only the fish in one meter stretches of the shore where the corresponding bit of the net is 1. Find the spot where you should place the left end of your net in order to catch the largest possible number of fish using an algorithm which runs in time O(n log n)

Step	Shoreline
1	$A = \langle A_0, A_1, \ldots, A_{100n-1} \rangle$
2 polynomial $\Rightarrow O(n)$	$P_A(x) = A_0 + A_1 x + A_2 x^2 + \ldots + A_{100n-1} x^{100n-1}$
3. Compute DFT using FFT $\Rightarrow O(n \log n)$	$\{P_A(1), P_A(\omega_{200n-1}^2), \ldots, P_A(\omega_{200n-1}^{200n-2})\}$;

Step	Net
1	$N = \langle N_0, N_1, \ldots, N_{n-1} \rangle$ However we will need to revert the recording of the net to correctly visualise the convolution $N' = \langle N'_0, N'_1, \ldots, N'_{n-1} \rangle$
2 polynomial $\Rightarrow O(n)$	$P_{N'}(x) = N'_0 + N'_1 x + N'_2 x^2 + \ldots + N'_{n-2} x^{n-1}$
3. Compute DFT using FFT $\Rightarrow O(n \log n)$	$\{P_{N'}(1), P_{N'}(\omega_{2n-1}), P_{N'}(\omega_{2n-1}^2), \ldots, P_{N'}(\omega_{2n-1}^{2n-2})\}$

Step	Shoreline and net
4. Multiply shoreline array and net array to retrieve convolution vector $P_C(x) = A * N'$ $\Rightarrow O(n)$	$\{P_A(1)P_{N'}(1), P_A(\omega_{200n-1})P_{N'}(\omega_{2n-1}), \ldots, P_A(\omega_{200n-1}^{200n-2})P_{N'}(\omega_{2n-1}^{2n-2})\}$
5. IDFT format $\Rightarrow O(n \log n)$	$P_C(x) = \sum_{j=0}^{200n-2} \left(\sum_{i=0}^{j} A_i N'_{j-i} \right) x^j$
	$C = \left\langle \sum A_i N'_{j-i} \right\rangle_{j=0}^{j=200n-2}$

The convolution performed will have an overall run time of $O(n \log n)$ and will be represented by $A * N' = P_C(x)$.

The above table shows how we use the shoreline array and fishnet array as convolution vectors with lengths 100n and n respectively. These arrays are multiplied together with a resulting array which will represent the number of fish that can be caught when casting the net between A [i - n: j]. Before multiplication however it is best to reverse the net array as the resultant vector will come out reversed if not.

The convolution multiplication will be achieved by first calculating the Discrete Fourier Transformation for both the shoreline array and net array. This will be done using Fast Fourier Transform (FFT) which runs in $O(n \log n)$ time

Next we will go through the resulting convolution vector C to find where result[i] is maximum. This point will be from the right side of the net and the left point will be i-n (as the net size is n). This will take $O(n)$ time and so the overall run time will remain $O(n \log n)$.