Algorithms Midterm Exam

Four questions, marked out of a total of $25 + 20 + 30 + 25 =$ **100 marks**.

Note: when we say "find an efficient algorithm" we mean "find an algorithm that runs in time $O(n^k)$ for some fixed integer k"; in these cases we are more interested in the correctness of your algorithm rather than in making it run in a specific time (for as long as its running time is polynomial in the size n of the input; brute force algorithms which run in an exponential time will bring you zero credit). If you use an algorithm covered in lectures you can just quote it; you do not need to describe its details.

---

1. (a) [**10 marks**] You are given an array A consisting of $2n-1$ integers. Design an algorithm which finds all of the n possible sums of n consecutive elements of A and **which runs in time $\mathbf{O(n)}$**. Thus, you have to find the values of all of the sums

$$\begin{aligned} S[1] &= A[1] + A[2] + \ldots + A[n-1] + A[n]; \\ S[2] &= A[2] + A[3] + \ldots + A[n] + A[n+1]; \\ &\vdots \qquad\quad \vdots \qquad\quad \vdots \\ S[n] &= A[n] + A[n+1] + \ldots A[2n-2] + A[2n-1], \end{aligned}$$

and your algorithm **should run in time $\mathbf{O(n)}$**.

   (b) [**15 marks**] You are a fisherman, trying to catch fish with a net that is W meters wide. Using your advanced technology, you know that the positions of all N fish in the sea can be represented as integers on a number line. There may be more than one fish at the same location.

   To catch the fish, you will cast your net at position x, and will catch all fish with positions between x and $x + W$, **inclusive**. Given N, W and an array $X[1..N]$ denoting the positions of fish in the sea, give an $\mathbf{O(N \log N)}$ algorithm to find the maximum number of fish you can catch by casting your net once.

   For example, if $N = 7, W = 3$ and $X = [1, 11, 4, 10, 6, 7, 7]$, then the most fish you can catch is 4: by placing your net at $x = 4$, you will catch one fish at position 4, one fish at position 6 and two fish at position 7.

2. (a) [**5 marks**] Compute by any method you wish the (linear) convolution $s * s$ of the sequence $s = \langle 1, 2, 0, 4 \rangle$ with itself. (Note that there is no requirement on the efficiency of your method, and that the sequence is really short!)

   (b) [**5 marks**] if a sequence x has n terms and sequence y has k terms, how many terms does the convolution $x * y$ of these two sequences have?

   (c) [**5 marks**] Is it true that $s * t = t * s$ for any two sequences s and t? Explain why or why not.

(d) **[5 marks]** Describe how we compute **efficiently** the convolution of two (long) sequences?

3. (a) **[20 marks]** Along the long, straight road from Loololong to Goolagong there are N houses and N hubs. Each hub is capable of providing an internet connection to **one house only**, and doing so requires installing a cable between the hub and the house it connects. Given two arrays $A[1..N]$ and $B[1..N]$ describing the locations of the houses and the hubs along the road, respectively, design an efficient algorithm that finds the **minimum total length** of cable required to connect every house to the internet.

   (b) **[10 marks]** Prove that your solution is optimal.

4. **[25 marks]** You and a friend find yourselves on a TV game show! The game works as follows. There is a **hidden** $N \times N$ table A. Each cell $A[i, j]$ of the table contains a single integer and no two cells contain the same value.

   At any point in time, you may ask the value of a single cell to be revealed.

   To win the big prize, you need to find the N cells each containing the **maximum** value in its row. Formally, you need to produce an array $M[1..N]$ so that $A[r, M[r]]$ contains the maximum value in row r of A, i.e., such that $A[r, M[r]]$ is the largest integer among $A[r, 1], A[r, 2], \ldots, A[r, N]$. In addition, to win, you should ask at most $\mathbf{O(N \log N)}$ many questions. For example, if the hidden grid looks like this:

	Column 1	Column 2	Column 3	Column 4
Row 1	**10**	5	8	3
Row 2	1	**9**	7	6
Row 3	-3	**4**	-1	0
Row 4	-10	-9	-8	**2**

   then the correct output would be $M = [1, 2, 2, 4]$.

   Your friend has just had a go, and sadly failed to win the prize because they asked N^2 many questions which is too many. However, they whisper to you a hint: they found out that M is **non-decreasing**. Formally, they tell you that $M[1] \leq M[2] \leq \cdots \leq M[N]$ (this is the case in the example above).

   Design an algorithm which asks at most $\mathbf{O(N \log N)}$ many questions that produces the array M correctly, even in the very worst case.

   Hint: Note that you do not have enough time to find out the value of every cell in the grid! Try determining $M[N/2]$ first, and see if divide-and-conquer is of any assistance.

END OF EXAM