1 Recurrence Use the recursion tree method to solve the following recurrence:

$$T(n) = \begin{cases} T(n/2) + 4T(n/4) + cn^2 & \text{for } n \geq 2 \\ c & \text{otherwise} \end{cases}$$

2 Chocolate Agency You are a consultant for a chocolate agency that buys (large) boxes of chocolate from producers and sells them to chocolate stores. The chocolate stores then uses the chocolate to make delicate filled chocolate they can sell. Consider a time period period of n days:

- On each day i there is exactly one producer that produces and sells the chocolate at a price of s_i.
- On each day j there is exactly one store willing to pay b_j to buy a fresh box of chocolate.
- The quality of the chocolate decreases each day after its prodcution day. Therefore, a store will only pay $b_j - 100\$$ for a box that is one day old, $b_j - 200\$$ for a box that is two days old, etc.

When matching a producer and a store, the agency earns the difference between the chocolate producer's selling price and the price the store buys it for. That is, the profit of the agency is $b_j - s_i - (j-i) \cdot 100\$$ for chocolate bought from a producer on day i and sold to a store on day j.

Your job is to devise an algorithm that decides on a pair of days i and j such that the agency maximizes their profit (obviously i must be at most j).

The input to the algorithm is n, the producers' sale prices $[s_1, s_2, \ldots, s_n]$, and the stores' buy prices $[b_1, b_2, \ldots, b_n]$.

2.1 Let $n = 6$, and consider the following example:

`days`	1	2	3	4	5	6
s_i	500\$	150\$	500\$	300\$	450\$	200\$
b_i	700\$	300\$	400\$	300\$	600\$	300\$

What is the maximum possible profit? Which pair of days gives the maixmum profit?

2.2 Given a day x describe an algorithm that in $O(n)$ time finds the best pair (i, j) where $i \leq x$ and $j > x$. The best pair is the pair that maximizes the profit. Remember to argue that your algorithm is correct.

2.3 Describe a divide-and-conquer algorithm that finds the pair of producers and stores that maximizes the profit. Remember to argue that your algorithm is correct.

2.4 Let $T(n)$ be the worst case running time of your algorithm. Give a recurrence for $T(n)$ (and explain why it is correct). What is the asymptotic running time of your algorithm (explain how you obtained the result)?