Collaboration is allowed, but you must write your own solutions. Proofs of correctness are a must.

Problem 1 (Randomized Algorithms)
Solve: [Kleinberg Tardos] Chapter 13, problem 1, page 782.

Problem 2 (Randomized Algorithms)

Hint: Consider an agent such that there are \( k \) agents with a higher bid than her. What is the probability that her bid results in an update of \( b^* \)?

Problem 3 (Randomized Algorithms)

Problem 4
Solve Problem 5 from the 2019 PhD exam:
https://cs.nyu.edu/home/phd/algorithms_exams/algorithms_2019fall_exam.pdf

Problem 5
In this problem, we explore the notion of oracle reducibility. If \( A \) is a language, then a Turing machine with oracle \( A \) is a Turing machine with a “magical” subroutine that decides membership in \( A \). In other words, the subroutine, when given a string \( w \), tells the machine whether or not \( w \in A \). Let
\[
\text{HALT}_{TM} = \{ \langle M, x \rangle \mid M \text{ is a Turing machine that halts on } x \}.
\]
Show that there is a Turing machine with oracle \( \text{HALT}_{TM} \) that decides the following problem with only two questions to the oracle: Given three (machine, input) pairs \( \langle M_1, x_1 \rangle, \langle M_2, x_2 \rangle, \langle M_3, x_3 \rangle \), decide for each pair whether the Turing machine halts on the corresponding input.

Note: This is trivial if one allows three questions. Just ask the oracle whether \( \langle M_i, x_i \rangle \in \text{HALT}_{TM} \) for \( i = 1, 2, 3 \).

Problem 6
Show that the collection of Turing-recognizable languages is closed under the operation of (a) union (b) concatenation (c) star and (d) intersection. What about complementation operation?