Problem 1 (0 points).

Solved Exercise 2 (page 779, *Algorithm Design* [KT]). (a) and (b) and their solutions are provided. (c) Design a 2-approximation algorithm for the general form described in (b).

Solution for (c). We can use the technique of *conditional expectation* to derandomize the randomized algorithm given in (b) to get a (deterministic) 2-approximation algorithm. Specifically, since $k = \mathbb{E}(X) = \mathbb{E}(X \mid x_1 = 1) \cdot \Pr(x_1 = 1) + \mathbb{E}(X \mid x_2 = 0) \cdot \Pr(x_1 = 0) = 0.5 \cdot \mathbb{E}(X \mid x_1 = 1) + 0.5 \cdot \mathbb{E}(X \mid x_1 = 0)$, we know that either $\mathbb{E}(X \mid x_1 = 1) \geq k/2$ or $\mathbb{E}(X \mid x_1 = 0) \geq k/2$. This suggests the following derandomization algorithm (similar to the one for MAX-3SAT problem):

```
Algorithm DERAND
    for $i = 1$ to $n$
        compute $A_1 = \mathbb{E}(X \mid x_1 = x_1^*, x_2 = x_2^*, \cdots, x_{i-1} = x_{i-1}^*, x_i = 1)$
        compute $A_0 = \mathbb{E}(X \mid x_1 = x_1^*, x_2 = x_2^*, \cdots, x_{i-1} = x_{i-1}^*, x_i = 0)$
        if $A_1 \geq A_0$
            set $x_i^* = 1$
        else
            set $x_i^* = 0$
        end if
    end for
    return $\{x_1^*, x_2^*, \cdots, x_n^*\}$
end DERAND
```

The above algorithm requires to compute the conditional expectation, i.e., A_1 and A_0. Clearly, $A_1 = \sum_{1 \leq r \leq k} \mathbb{E}(X_r \mid x_1 = x_1^*, x_2 = x_2^*, \cdots, x_{i-1} = x_{i-1}^*, x_i = 1)$. Consider the r-th equation. Case 1: all its variables have been assigned, i.e., they are all in $\{x_1, x_2, \cdots, x_i\}$; in this case we can verify whether the equation is true or not: if it is true we have $\mathbb{E}(X_r \mid x_1 = x_1^*, x_2 = x_2^*, \cdots, x_{i-1} = x_{i-1}^*, x_i = 1) = 1$, and if it is not true we have $\mathbb{E}(X_r \mid x_1 = x_1^*, x_2 = x_2^*, \cdots, x_{i-1} = x_{i-1}^*, x_i = 1) = 0$. Case 2: not all its variables has been assigned; in this case we have $\mathbb{E}(X_r \mid x_1 = x_1^*, x_2 = x_2^*, \cdots, x_{i-1} = x_{i-1}^*, x_i = 1) = 0.5$, using the same argument in (b).

Problem 2 (0 points).

Exercise 7 (page 787, *Algorithm Design* [KT]).

Solution. (a) We have proved that $\mathbb{E}(Z) = k/2$. An tight example (with 1 variable and 2 clauses) is as follows. $C_1 : x_1$ and $C_2 : \overline{x_1}$.

(b) Let A be the set of single-literal clauses. For each clause in A we independently at random satisfy it with probability $p \geq 0.5$ (we will determine the actual value of p later on). For examples, if such a clause is x_2 then we set $x_2 = 1$ with probability p; if such a clause is $\overline{x_4}$ then we set $x_4 = 0$ with probability p. For all other variables that have not been assigned, we independently at random set them to 1 with probability of exactly 1/2.

We now analyze this algorithm. For each clause C in A, clearly $\Pr(C = 1) = p$. For each clause C that is not in A, we know that C contains at least 2 literals. The worst case is that C contains exactly 2 literals for which their opporsite literals have been assigned in processing A. (For example, C is $\overline{x_2} \vee x_4$, and x_2 and $\overline{x_4}$ are two single-literal clauses in A.) In this case, $\Pr(C = 1) = 1 - \Pr(C = 0) = 1 - p^2$. This is the worse case. In other words, for any clause C that not in A we must have that $\Pr(C = 1) \geq 1 - p^2$.

To get the actual value of p, we simply let $p = 1 - p^2$, i.e., let each clause have the same probability of being satisfied. This gives $p = 0.62$. Hence, we have $\mathbb{E}(Z) = 0.62k$.

(c) For each pair of single-literal clauses x and $\bar{x}$, we simply keep one of them and discard the other. Let k_1 be the number of removed clauses in this way. We then run the algorithm given in (b) on the remaining $k - k_1$ clauses.

We now analyze this algorithm. According to (b), the expected number of satisfied clauses equals to $0.62 \cdot (k - k_1)$. On the other side, consider the optimal solution. Note that for any pair of conflicting clauses, at most one of them can be satisfied in any assignment. Therefore, the maximized number of clauses that can be satisfied is $(k - k_1)$. Combined, in expectation, this randomized algorithm gives at least 0.62 fraction of the satisfied clauses in optimal solution.