CSE 4190.101: Discrete Mathematics	Fall 2021

Problem Set 4 Solution

Instructor: Yongsoo Song — **Due on:** Dec 10, 2021

Please submit your answer through eTL. It should be a single PDF file (not jpg), either typed or scanned. **Please include your student ID and name (e.g. 2020-12345 YongsooSong.pdf).** You may discuss with other students the general approach to solve the problems, but the answers should be written in your own words.

- You should cite any reference that you used, and mention what you used it for.
- The reference information should be specific so that TAs are able to find the exact material you used. For example, it is not allowed to simply mention that "I referred a lecture note of Discrete Mathematics class at * university".
- Similarly, if your reference includes a url, type it or submit a separate text file (instead of handwritten address) so that TAs can easily visit the page.
- All references should be publicly accessible. Otherwise, attach the reference to your submission.

Overall Grading Policy

1. No submissions or late submissions. (0 points received.)
2. Cheating detected. (0 points received.)
3. Not a single file. (including multiple files compressed into a single ZIP file) (10 points deducted.)
4. Not in a PDF format. (10 points deducted.)
5. Necessary citations missing. (Several points may be deducted according to severity.)

Problem 1 (15 points)

Suppose that $\{a_n\}$ satisfies the linear nonhomogeneous recurrence relation

$$a_n = c_1 \cdot a_{n-1} + c_2 \cdot a_{n-2} + \cdots + c_k \cdot a_{n-k} + F(n),$$

where $c_1, c_2, \ldots, c_k$ are real numbers, and

$$F(n) = (b_t \cdot n^t + b_{t-1} \cdot n^{t-1} + \cdots + b_0) \cdot s^n$$

where $b_0, b_1, \ldots b_t$ and s are real numbers. Show that the following statements are true:

1. If s is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, then there is a particular solution of the form

$$(p_t \cdot n^t + p_{t-1} \cdot n^{t-1} + \cdots + p_0) \cdot s^n$$

.

Let's use proof by induction.

let P(t) be "When s is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, then there is a particular solution of the form $(p_t \cdot n^t + p_{t-1} \cdot n^{t-1} + \cdots + p_0) \cdot s^n$.

Base case ($t = 0$)
$a_n = c_1 \cdot a_{n-1} + c_2 \cdot a_{n-2} + \cdots + c_k \cdot a_{n-k} + F(n)$
For $a_n = p_0 \cdot s^n$,
$p_0 \cdot s^n = c_1 \cdot p_0 \cdot s^{n-1} + c_2 \cdot p_0 \cdot s^{n-2} + \cdots + c_k \cdot p_0 \cdot s^{n-k} + b_0 \cdot s^n$
$p_0 \cdot (s^n - c_1 \cdot s^{n-1} - c_2 \cdot s^{n-2} - \cdots - c_k \cdot s^{n-k}) = b_0 \cdot s^n$
However, since s is not a root of the characteristic equation, $s^n - c_1 \cdot s^{n-1} - c_2 \cdot s^{n-2} - \cdots - c_k \cdot s^{n-k} \neq 0$
Take $p_0 = \frac{b_0 \cdot s^n}{s^n - c_1 \cdot s^{n-1} - c_2 \cdot s^{n-2} - \cdots - c_k \cdot s^{n-k}}$
Thus, there exists a particular solution of the recurrence relation of the form $a_n = p_0 \cdot s^n$.

Now assume P(i) is true, and show that P(i+1) still holds.
By inductive assumption, there is a particular solution $(p_i \cdot n^i + p_{i-1} \cdot n^{i-1} + \cdots + p_0) \cdot s^n$
This means there exits $p_0, p_1, \ldots p_i$ that satisfies

$$\begin{gathered}(p_i \cdot n^i + p_{i-1} \cdot n^{i-1} + \cdots + p_0) \cdot s^n \\ = c_1 \cdot (p_i \cdot (n-1)^i + p_{i-1} \cdot (n-1)^{i-1} + \cdots + p_0) \cdot s^{n-1} \\ + c_2 \cdot (p_i \cdot (n-2)^i + p_{i-1} \cdot (n-2)^{i-1} + \cdots + p_0) \cdot s^{n-2} \\ \cdots \\ + c_k \cdot (p_i \cdot (n-k)^i + p_{i-1} \cdot (n-k)^{i-1} + \cdots + p_0) \cdot s^{n-k} \\ + (b_i \cdot n^i + b_{i-1} \cdot n^{i-1} + \cdots + b_0) \cdot s^n\end{gathered}$$

Assume there exists a particular solution of the form $(p_{i+1} \cdot n^{i+1} + p_i \cdot n^i + \cdots + p_0) \cdot s^n$
Substituting this results in the following.

$$\begin{gathered}(p_{i+1} \cdot n^{i+1} + p_i \cdot n^i + \cdots + p_0) \cdot s^n \\ = c_1 \cdot (p_{i+1} \cdot (n-1)^{i+1} + p_i \cdot (n-1)^i + \cdots + p_0) \cdot s^{n-1} \\ + c_2 \cdot (p_{i+1} \cdot (n-2)^{i+1} + p_i \cdot (n-2)^i + \cdots + p_0) \cdot s^{n-2} \\ \cdots \\ + c_k \cdot (p_{i+1} \cdot (n-k)^{i+1} + p_i \cdot (n-k)^i + \cdots + p_0) \cdot s^{n-k} \\ + (b_{i+1} \cdot n^{i+1} + b_i \cdot n^i + \cdots + b_0) \cdot s^n\end{gathered}$$

From the inductive hypothesis, it is possible to find $p_0 \ldots p_i$ that cancels out and leaves
$p_{i+1} \cdot n^{i+1} \cdot s^n = c_1 \cdot p_{i+1} \cdot (n-1)^{i+1} \cdot s^{n-1} + c_2 \cdot p_{i+1} \cdot (n-2)^{i+1} \cdot s^{n-2} \ldots c_k \cdot p_{i+1} \cdot (n-k)^{i+1} \cdot s^{n-k} + b_{i+1} \cdot n^{i+1} \cdot s^n$
Rewriting this becomes
$p_{i+1} \cdot (n^{i+1} \cdot s^n - c_1 \cdot (n-1)^{i+1} \cdot s^{n-1} - c_2 \cdot (n-2)^{i+1} \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^{i+1} \cdot s^{n-k}) = b_{i+1} \cdot n^{i+1} \cdot s^n$
Since s is not a root, $n^{i+1} \cdot s^n - c_1 \cdot (n-1)^{i+1} \cdot s^{n-1} - c_2 \cdot (n-2)^{i+1} \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^{i+1} \cdot s^{n-k} \neq 0$
Thus, $p_{i+1} = \frac{b_{i+1} \cdot n^{i+1} \cdot s^n}{n^{i+1} \cdot s^n - c_1 \cdot (n-1)^{i+1} \cdot s^{n-1} - c_2 \cdot (n-2)^{i+1} \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^{i+1} \cdot s^{n-k}}$ exists.

2. If s is a root of this characteristic equation and its multiplicity is m, then there is a particular solution of the form

$$n^m \cdot (p_t \cdot n^t + p_{t-1} \cdot n^{t-1} + \cdots + p_0) \cdot s^n$$

Let's use proof by induction.

let P(t) be "If s is a root of this characteristic equation and its multiplicity is m, then there is a particular solution of the form $n^m \cdot (p_t \cdot n^t + p_{t-1} \cdot n^{t-1} + \cdots + p_0) \cdot s^n$.

Base case $(t = 0)$
$a_n = c_1 \cdot a_{n-1} + c_2 \cdot a_{n-2} + \cdots + c_k \cdot a_{n-k} + F(n)$
For $a_n = n^m \cdot p_0 \cdot s^n$,
$n^m \cdot p_0 \cdot s^n = c_1 \cdot (n-1)^m \cdot p_0 \cdot s^{n-1} + c_2 \cdot (n-2)^m \cdot p_0 \cdot s^{n-2} + \cdots + c_k \cdot (n-k)^m \cdot p_0 \cdot s^{n-k} + b_0 \cdot s^n$
$p_0 \cdot (n^m s^n - c_1 \cdot (n-1)^m \cdot s^{n-1} - c_2 \cdot (n-2)^m \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^m \cdot s^{n-k}) = b_0 \cdot s^n$
However, since s is the root of the characteristic equation of multiplicity m,
$n^j s^n - c_1 \cdot (n-1)^j \cdot s^{n-1} - c_2 \cdot (n-2)^j \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^j \cdot s^{n-k} \neq 0$ for all $j \geq m$
Take $p_0 = \frac{b_0 \cdot s^n}{n^m s^n - c_1 \cdot (n-1)^m \cdot s^{n-1} - c_2 \cdot (n-2)^m \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^m \cdot s^{n-k}}$
Thus, there exists a particular solution of the recurrence relation of the form $a_n = n^m \cdot p_0 \cdot s^n$.

Now assume P(i) is true, and show that P(i+1) still holds.
By inductive assumption, there is a particular solution $n^m \cdot (p_i \cdot n^i + p_{i-1} \cdot n^{i-1} + \cdots + p_0) \cdot s^n$
This means there exits $p_0, p_1, \ldots p_i$ that satisfies

$$\begin{gathered}
n^m \cdot (p_i \cdot n^i + p_{i-1} \cdot n^{i-1} + \cdots + p_0) \cdot s^n \\
= c_1 \cdot (n-1)^m \cdot (p_i \cdot (n-1)^i + p_{i-1} \cdot (n-1)^{i-1} + \cdots + p_0) \cdot s^{n-1} \\
+ c_2 \cdot (n-2)^m \cdot (p_i \cdot (n-2)^i + p_{i-1} \cdot (n-2)^{i-1} + \cdots + p_0) \cdot s^{n-2} \\
\cdots \\
+ c_k \cdot (n-k)^m \cdot (p_i \cdot (n-k)^i + p_{i-1} \cdot (n-k)^{i-1} + \cdots + p_0) \cdot s^{n-k} \\
+ (b_i \cdot n^i + b_{i-1} \cdot n^{i-1} + \cdots + b_0) \cdot s^n
\end{gathered}$$

Assume there exists a particular solution of the form $n^m \cdot (p_{i+1} \cdot n^{i+1} + p_i \cdot n^i + \cdots + p_0) \cdot s^n$
Substituting this results in the following.

$$\begin{gathered}
n^m \cdot (p_{i+1} \cdot n^{i+1} + p_i \cdot n^i + \cdots + p_0) \cdot s^n \\
= c_1 \cdot (n-1)^m \cdot (p_{i+1} \cdot (n-1)^{i+1} + p_i \cdot (n-1)^i + \cdots + p_0) \cdot s^{n-1} \\
+ c_2 \cdot (n-2)^m \cdot (p_{i+1} \cdot (n-2)^{i+1} + p_i \cdot (n-2)^i + \cdots + p_0) \cdot s^{n-2} \\
\cdots \\
+ c_k \cdot (n-k)^m \cdot (p_{i+1} \cdot (n-k)^{i+1} + p_i \cdot (n-k)^i + \cdots + p_0) \cdot s^{n-k} \\
+ (b_{i+1} \cdot n^{i+1} + b_i \cdot n^i + \cdots + b_0) \cdot s^n
\end{gathered}$$

From the inductive hypothesis, it is possible to find $p_0 \ldots p_i$ that cancels out and leaves
$p_{i+1} \cdot n^{i+1+m} \cdot s^n = c_1 \cdot p_{i+1} \cdot (n-1)^{i+1+m} \cdot s^{n-1} + c_2 \cdot p_{i+1} \cdot (n-2)^{i+1+m} \cdot s^{n-2} \ldots c_k \cdot p_{i+1} \cdot (n-k)^{i+1+m} \cdot s^{n-k} + b_{i+1} \cdot n^{i+1} \cdot s^n$

Rewriting this becomes
$p_{i+1} \cdot (n^{i+1+m} \cdot s^n - c_1 \cdot (n-1)^{i+1+m} \cdot s^{n-1} - c_2 \cdot (n-2)^{i+1+m} \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^{i+1+m} \cdot s^{n-k}) = b_{i+1} \cdot n^{i+1} \cdot s^n$
Since $i+1+m \geq m$,
$n^{i+1+m} \cdot s^n - c_1 \cdot (n-1)^{i+1+m} \cdot s^{n-1} - c_2 \cdot (n-2)^{i+1+m} \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^{i+1+m} \cdot s^{n-k} \neq 0$
Thus, $p_{i+1} = \frac{b_{i+1} \cdot n^{i+1} \cdot s^n}{n^{i+1+m} \cdot s^n - c_1 \cdot (n-1)^{i+1+m} \cdot s^{n-1} - c_2 \cdot (n-2)^{i+1+m} \cdot s^{n-2} - \cdots - c_k \cdot (n-k)^{i+1+m} \cdot s^{n-k}}$ exists.

Problem 2 (10 points)

Suppose that each person in a group of n people votes for exactly two people from a slate of candidates to fill two positions on a committee. The top two finishers both win positions as long as each receives more than $n/2$ votes.

1. Devise a divide-and-conquer algorithm that determines whether the two candidates who received the most votes each received at least $n/2$ votes and, if so, determine who these two candidates are.
   basic step. if n = 1: then the two people on the list are winners.
   recursive step. if there are n voters: then we can assume that the list of all possible names contains $2n$ names. We divide the lists into two sublists (each) until we obtain sublists with at most 3 names (names that are repeated twice will then have a majority in the sublist). When comparing the two lists of n names, then there are at most 3 names with a majority in each list. We need to compare the 3 names in one list with the 3 names in the other list as well as the frequency of occurrence of the names, which requires at most 12n comparisons. in total, we then note that the number of comparisons is twice the number of comparisons in a sublist of half length increased by at most 12n comparions $f(n) = 2f(n/2) + 12n$ ref. Quizlet

2. Use the master theorem to give a big-O estimate for the number of comparisons needed by the algorithm you devised.
   Master theorem When $f(n) = af(n/b) + cn^b$ then,
   if $a < b^d$: $f(n) = O(n^d)$,
   if $a = b^d$: $f(n) = O(n^d \log n)$,
   if $a > b^d$:$f(n) = O(n^{\log_b a})$
   here, a = 2, b = 2, c = 12, d = 1
   So,
   $$O(n \log n)$$

Problem 3 (15 points)

Suppose someone picks a number x from a set of n numbers. A second person tries to guess the number by successively selecting subsets of the n numbers and asking the first person whether x is in each set. The first person answers either "yes" or "no". Ulam's problem, proposed by Stanislaw Ulam in 1976, asks for the number of queries required to find x, supposing that the first person is allowed to lie exactly once.

1. Show that by dividing the initial set of n elements into four parts, each with $n/4$ elements, $1/4$ of the elements can be eliminated using two queries. [Hint: Use two queries, where each of the queries asks whether the element is in the union of two of the subsets with $n/4$ elements and where one of the subsets of $n/4$ elements is used in both queries.]
   Following the hint, suppose that I divide the initial sets into four subsets with n/4 elements: A, B, C and D. Without loss of generality, I make two questions: (1) whether the element x is in $A \cup B$ and (2) whether the element x is in $A \cup B$. There are four possibilities for the answer to these questions:

(a) If yes is answered to both questions, then x must be in $A \cup B$ or $A \cup C$ (even if only one answer is truthful). Thus, I can be certain that x is not in D. Therefore, D can be eliminated using two queries.

(b) If no is answered to both questions, then x is not in $A \cup B$ or not in $A \cup C$. Thus, it is certainly not in A, so A can be eliminated

(c) If yes is answered to question (1) and no is answered to question (2), with at most one lie, several possibilities arise. Analyzing the possibilities below, we conclude that C can always be eliminated in this case:
If both answers are truthful, then x must not be in $A \cup C$; thus, A or C could be eliminated;
If the first answer is a lie, then x is not in $A \cup B$ and not in $A \cup C$; therefore, A, B and C could be eliminated;
If the second answer is a lie, then x is in A; therefore, B, C or D could be eliminated.

(d) If no is answered to question (1) and yes is answered to question (2), by an analogous reasoning, B can always be eliminated in this case.

2. Show that if $f(n)$ equals the number of queries used to solve Ulam's problem using the method above, then $f(n) = f(3n/4) + 2$ when n is divisible by 4. Solve the recurrence relation.
f n is divisible by four, I can solve this problem by asking 2 questions to eliminate n/4 elements of the initial set, and then ask f(3n/4) questions for the rest of the 3n/4 elements. Therefore:

$$f(n) = 2 + f(3n/4)$$

Problem 4 (10 points)

How many permutations of the 26 letters of the English alphabet do not contain any of the strings *fish*, *rat* or *bird*?

Total number of permutations = 26!
Number of permutations containing a string *fish*: $= (26 - 4 + 1)! = 23!$ (*fish* is treated as a single word, and rule out 4 letters *f*, *i*, *s*, *h* from 26 letters)
Number of permutations containing a string *rat*: $= (26 - 3 + 1)! = 24!$
Number of permutations containing a string *bird*: $= (26 - 4 + 1)! = 23!$
Number of permutations containing strings *fish* and *rat*: $= (26 - 4 - 3 + 2)! = 21!$
Number of permutations containing strings *fish* and *bird*: 0 (i cannot appear twice)
Number of permutations containing strings *rat* and *bird*: 0 (r cannot appear twice)
Number of permutations containing strings *fish*, *rat*, and *bird*: 0 (i or r cannot appear twice)
Number of permutations containing strings *fish* or *rat* or *bird* $= 23! + 24! + 23! - 21! - 0 - 0 + 0 = 24! + 2 \cdot 23! - 21!$
Answer $= 26! - (24! + 2 \cdot 23! - 21!) = 26! - 24! - 2 \cdot 23! + 21!$
(5 points off for each calculation mistake.)

Problem 5 (20 points)

In this exercise we construct a dynamic programming algorithm for solving the problem of finding a subset S of items chosen from a set of n items where item i has a weight w_i, which is a positive integer, so that the total weight of the items in S is a maximum but does not exceed a fixed weight limit W. Let $M(j, w)$ denote the maximum total weight of the items in a subset of the first j items such that this total weight does not exceed w. This problem is known as the knapsack problem.

1. Show that
$$M(j,w) = \begin{cases} M(j-1,w) & \text{if } w_j > w, \\ \max\{M(j-1,w), w_j + M(j-1, w-w_j)\} & \text{otherwise.} \end{cases}$$

   (5 points, no partial credits) When $w_j > w$, we cannot add the jth item since the weight of the jth item itself already exceeds the given w. In this case, $M(j,w)$ is equal to the maximum total weight of the first $(j-1)$ items, or $M(j-1,w)$.
   When $w_j \leq w$, there are two possible cases. First, we can choose to add the jth item. Adding the jth item implies that there is enough room for this item, even after adding candidates from the first $(j-1)$ items. Therefore, the maximum total weight of the items before adding the jth item can be denoted as $M(j-1, w-w_j)$. After adding the jth item, we can denote $M(j,w)$ as $w_j + M(j-1, w-w_j)$. Second, we can choose not to add the jth item. In this case, it is equivalent to the case where $M(j,w)$ is equal to the maximum total weight of the first $j-1$ items, or $M(j-1,w)$. At last, we must choose the one with the bigger value among the two possible cases. Therefore, $M(j,w) = \max\{M(j-1,w), w_j + M(j-1,w-w_j)\}$.

2. Construct a dynamic programming algorithm for determining the maximum total weight of items so that this total weight does not exceed W.

   (10 points) Answered together with subproblem 3 in below **Algorithm 1**.
   (8 points off for an algorithm that does not use dynamic programming.)

3. Explain how you can use the values $M(j,w)$ to find a subset of items with maximum total weight not exceeding W.

   (5 points, no partial credits) Below algorithm shows how a dynamic programming algorithm for subproblem 2 works, and lines in blue are added to find a subset of items for subproblem 3.

Algorithm 1 Dynamic Programming Algorithm

input: $w_1, w_2, \cdots, w_n$: weights of each item, W: a fixed weight limit
for $w \leftarrow 0$ to W **do**
    $M[0,w] \leftarrow 0$ ▷ Initialize M for the maximum total weight.
    $S[0,w] \leftarrow false$ ▷ Initialize S for a subset of items.
end for
for $j \leftarrow 1$ to n **do**
    for $w \leftarrow 0$ to W **do**
        if $w_j > w$ **then**
            $M[j,w] \leftarrow M[j-1,w]$
            $S[j,w] \leftarrow false$ ▷ The jth item is not added.
        else ▷ $w_j \leq w$
            if $M[j-1,w] > w_j + M[j-1,w-w_j]$ **then**
                $M[j,w] \leftarrow M[j-1,w]$
                $S[j,w] \leftarrow false$ ▷ The jth item is not added.
            else ▷ $w_j \leq w$ and $M[j-1,w] \leq w_j + M[j-1,w-w_j]$
                $M[j,w] \leftarrow w_j + M[j-1,w-w_j]$
                $S[j,w] \leftarrow true$ ▷ The jth item is added.
            end if
        end if
    end for
end for
return $M[n,W]$ ▷ For subproblem 2.
$R \leftarrow W$

```
l ← empty list
for j ← n to 1 do
    if S[j][R] = true then
        l.add(j)
        R = R − w_j
    end if
end for
return l                ▷ For subproblem 3. l holds the indices of the items included in the subset.
```

Problem 6 (10 points)

Let R be a relation on a set A and $R^{-1} = \{(b, a) \mid (a, b) \in R\}$ be the inverse relation. Show that R is antisymmetric if and only if $R \cap R^{-1}$ is a subset of the diagonal relation $\Delta = \{(a, a) \mid a \in A\}$.

(10 points if logically correct. No partial credits.)

$$
\begin{aligned}
(a, b) \in R \cap R^{-1} &\iff (a, b) \in R \wedge (a, b) \in R^{-1} && \\
&\iff (a, b), (b, a) \in R && \text{(Def. of } R^{-1}\text{)} \\
\\
\therefore R \text{ is antisymmetric} &\iff (a, b), (b, a) \in R \longrightarrow a = b && \text{(Def. of antisymmetry)} \\
&\iff (a, b) \in R \cap R^{-1} \longrightarrow a = b && \forall a, b \in A \\
&\iff (a, b) \in R \cap R^{-1} \longrightarrow (a, b) \in \Delta && \text{(Def. of } \Delta\text{)} \\
&\iff R \cap R^{-1} \subseteq \Delta &&
\end{aligned}
$$

Problem 7 (10 points)

Do we necessarily get an equivalence relation when we form the symmetric closure of the reflexive closure of the transitive closure of a relation?

No we do not get an equivalence relation.The below is a counterexample
Consider a relation $R = \{(a, b), (c, b)\}$ on a set $S = \{a, b, c\}$
Transitive closure $R' = \{(a, b), (c, b)\}$
Reflexive closure $R'' = \{(a, b), (c, b), (a, a), (b, b), (c, c)\}$
Symmetric closure $R''' = \{(a, b), (b, a), (c, b), (b, c), (a, a), (b, b), (c, c)\}$

In order for this to be an equivalence relation, it needs to be reflexive, symmetric, and transitive.
However, for R''', $(a, b) \in R'''$ and $(b, c) \in R'''$, but $(a, c) \notin R'''$
Therefore it is not transitive, and thus it is not an equivalence relation.

Problem 8 (10 points)

Show that the property that a graph is bipartite is an isomorphic invariant.

(10 points if the proof is logically correct and contains clear construction of bipartition)

Given isomorphic graphs $G = (V, E), G' = (V', E')$ and the isomorphism f that maps V to V', let's assume that G is an bipartite graph with bipartition (V_1, V_2). We can consider two subsets V_1', V_2' which are corresponded to V_1, V_2 of V by f, or mathematically speaking, $V_1' = f(V_1), V_2' = f(V_2)$. Let's show that V_1', V_2' actually is a bipartition for G'.

By definition of graph isomorphism following simply holds:

$$\begin{aligned} (a,b) \in E &\iff (f(a), f(b)) \in E' \\ (a,b) \in V_a \times V_b &\iff (f(a), f(b)) \in f(V_a) \times f(V_b) \qquad\qquad \forall\, V_a, V_b \subseteq V \end{aligned}$$

Since G is a bipartite graph, following holds:

$$(a,b) \in E \implies (a,b) \in V_1 \times V_2 \text{ or } (a,b) \in V_2 \times V_1$$

Combining two results above, for all $a', b' \in V'$

$$\begin{aligned} (a', b') \in E' &\implies (a,b) \in E \qquad\qquad (a = f^{-1}(a'), b = f^{-1}(b')) && (1) \\ &\implies (a,b) \in V_1 \times V_2 \text{ or } (a,b) \in V_2 \times V_1 && (2) \\ &\implies (a', b') \in V_1' \times V_2' \text{ or } (a', b') \in V_1' \times V_2' && (3) \end{aligned}$$

Since V_1, V_2 are disjoint sets that covers V, V_1', V_2' also are disjoint sets that covers V'. Finally, we can conclude that V_1', V_2' is a bipartition from (3), and thus G' is a bipartite graph. We have shown that of two isomorphic graphs if one is a bipartite graph then other one is also a bipartite graph, and this implies that whether a graph is bipartite is isomorphic invariant.