Instructions Follow instructions carefully, failure to do so may result in points being deducted. Clearly label each problem and submit the answers in order. Staple this cover page to the front of your assignment for easier grading. Be sure to show sufficient work to justify your answer(s). If you are asked to prove something, you must give as formal, rigorous, and complete proof as possible. The CSE academic dishonesty policy is in effect (see http://www.cse.unl.edu/undergrads/academic_integrity.php). You are highly encouraged to typeset your homework using \LaTeX; if your answers are not legible, you may be required to use \LaTeX in future assignments. Print out a copy of the cover sheet and include it with your homework.

Partner Policy You may work in groups of at most two students. This is optional and you may work alone if you wish. If you opt to work in pairs, you must follow these guidelines:

1. You must work on all problems together. You may not simply partition the work between you.
2. You should not discuss problems with other groups or individuals beyond general questions.
3. Hand in only one hard copy (and possibly soft copy) under the first author’s name/cse login.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>150</td>
<td></td>
</tr>
</tbody>
</table>
1. 2 points (Rosen 6.1.4) A particular brand of shirt comes in 12 colors, has a male version and a female version, and comes in three sizes for each sex. How many different types of this shirt are made?

2. 10 points A pizza restaurant allows you to order a pizza with a certain number of toppings. In total, there are 12 distinct toppings to choose from.
   (a) Special 1 allows you to choose any 3 distinct toppings. How many possible pizzas are there?
   (b) Special 2 allows you to choose any 3 distinct toppings, but you must specify an ordering of the toppings as they are layered with special types of cheeses. How many possible pizzas are there?
   (c) Special 3 allows you to choose any 4 toppings, but you may request the same topping more than once (double, triple or quadruple the jalapeños for example). How many possible pizzas are there?
   (d) Special 4 allows you to choose up to 3 distinct toppings (you may choose none, one or two). How many possible pizzas are there?

3. 10 points Recall that the standard Mameluke card deck contains 52 cards (2–9, Jack, Queen, King, Ace in 4 suits).
   (a) A “four-of-a-kind” is a 5 card hand that contains the same card value in all four suits (four aces, four 2s, etc.; the fifth card may be anything). How many ways are there to form a 4 of a kind?
   (b) A “flush” is a 5 card hand that consists of any 5 cards but of the same suit. How many ways can you form a flush?
   (c) Bridge is a 4 player game played with a 52 card deck such that each player is dealt 13 cards each. How many different unique bridge hands are there (of 13 cards)?
   (d) Texas Hold ’em is a poker variation played with a standard 52 card deck with 2–9 players. Each player is dealt 2 cards and (eventually) 5 “community cards” are chosen. How many total unique deals are there when 5 players are involved?

4. 12 points (Rosen 6.1.42) A DNA sequence is a string consisting of some combination of four proteins, A, G, C, T. How many four element DNA sequences
   (a) do not contain the base T?
   (b) contain the sequence ACG?
   (c) contain all four bases A, T, C, and G?
   (d) contain exactly three of the four bases, A, T, C, and G?

5. 2 points (Rosen 6.2.16) How many numbers must be selected from the set \{1, 3, 5, 7, 9, 11, 13, 15\} to guarantee that at least one pair of these numbers add up to 16?

6. 2 points How many n-bit strings contain at least 1 zero?

7. 2 points How many bit strings of length 12 contain an equal number of 0s and 1s?

8. 4 points (Rosen 6.4.30) Seven women and nine men are on the faculty in the mathematics department at a school.
   (a) How many ways are there to select a committee of five members of the department if at least one woman must be on the committee?
   (b) How many ways are there to select a committee of five members of the department if at least one woman and at least one man must be on the committee?

9. 4 points (Rosen 8.5.2) There are 345 students at a college who have taken a course in calculus, 212 who have taken a course in discrete mathematics, and 188 who have taken courses in both calculus and discrete mathematics. How many students have taken a course in either calculus or discrete mathematics?
10. **6 points** (Rosen 8.5.6) Find the number of elements in \( A_1 \cup A_2 \cup A_3 \) if there are 100 elements in \( A_1 \), 1000 in \( A_2 \), and 10,000 in \( A_3 \) if

(a) \( A_1 \subseteq A_2 \) and \( A_2 \subseteq A_3 \)

(b) The sets are pairwise disjoint

(c) There are two elements common to each pair of sets and one element in all three sets.

11. **6 points** Let \( A = \{1, 2, \ldots, 20\} \).

(a) In how many ways can we partition \( A \) into two disjoint subsets?

(b) How many ways can we partition \( A \) into two disjoint subsets of equal size?

12. **8 points** Consider an \( n \times n \) grid of cells (indexed 0, \ldots, \( n-1 \)). Say we want to draw a path from the lower left cell \((1,1)\) to the upper right cell, \((n-1,n-1)\) by travelling only east or north cell-by-cell (no back-tracking is allowed, so we cannot travel west or south). How many different paths are there in terms of \( n \)? For example, for \( n = 4 \), the number of paths from \((0,0)\) to \((3,3)\) is 20 (see Figure 1).

![Grid Example](image)

(a) At a particular cell \((i,j)\), how many ways are there of reaching this point in terms of adjacent cells?

(b) How many paths are there for \( n = 5 \)?

(c) How many paths are there for \( n = 10 \)?

(d) Give a general formula in terms of \( n \) for the number of paths.

13. **6 points** A graph \( G = (V, E) \) is bipartite if its vertices can be partitioned into two subsets \( V = A \cup B \) such that all edges connect only vertices between the two sets (no two edges in the same set are connected).

(a) Prove or disprove: If a graph \( G \) is bipartite, then \( G \) is a tree.

(b) Prove or disprove: If a graph \( G \) is a tree, then it is bipartite.

14. **4 points** Prove or disprove: for all even \( n \geq 4 \), the complete graph \( K_n \) is bipartite.

15. **16 points** (Rosen 10.2.42) A sequence of integers \( d_1, d_2, \ldots, d_n \) is called graphic if it is the degree sequence of a simple undirected graph. Determine whether each of these sequences is graphic. For those that are, draw a graph having the given sequence. For those that are not, provide some reason for why no graph has such a sequence.
16. 6 points Let \( G = (V, E) \) be an undirected graph.
   
   (a) Prove or disprove: If \( |E| \leq |V| - 1 \) then \( G \) is acyclic.
   
   (b) \( G \) is said to be connected if every pair of vertices \( u, v \in V \) is connected by some path. How big does the edge set \( E \) have to be to guarantee that it is connected? (hint: its much larger than \( n \)).

17. 50 points Programming Problems

There are two programming problems available: Huffman Coding and Graph Isomorphism; for this problem you are required to do at least one of them. It is your choice as to which one you do. Moreover, you can choose to do both as potential bonus points (up to 50 points).

**Program 1: Graph Isomorphism Testing**

Two undirected graphs \( G_1 = (V_1, E_1) \), \( G_2 = (V_2, E_2) \) are said to be isomorphic, denoted \( G_1 \simeq G_2 \) if there exists a bijective mapping \( f : V_1 \to V_2 \) such that for every pair of vertices \( u, v \in V_1 \), \((u, v) \in E_1 \iff (f(u), f(v)) \in E_2 \).

An example of two isomorphic graphs can be found in Figure 2. These graphs are isomorphic as witnessed by the mapping \( v_1 \to v_2, v_2 \to v_4, v_3 \to v_3, v_4 \to v_1 \).

![Figure 2: Two isomorphic graphs. \( G_1 \simeq G_2 \)](image)

The **Graph Isomorphism Problem** is as follows: given two graphs \( G_1, G_2 \) determine whether or not \( G_1 \simeq G_2 \). You will write a program to solve the graph isomorphism problem for any two graphs. You have two options: you can write your program in Java using a nice graph library, JgraphT (relieving you of the need to write your own graph classes and input file parser), or you may write it any programming language (including Java) you wish (but then you will need to write a parser to handle the input/output).

**Java Program using JgraphT**

You will implement a static Java method, `getIsomorphicMap` in the Java source file provided on the course webpage. The method is detailed below.

```java
/**
 * If the given graphs are isomorphic, returns a map between
 * the vertex sets witnessing the isomorphism. If the given
```
* graphs are not isomorphic, returns null.

*/
public static Map<String, String> getIsomorphicMap(
    UndirectedGraph<String, DefaultEdge> a,
    UndirectedGraph<String, DefaultEdge> b) {
    // TODO: implement
    return null;
}

Note that this method uses a Java graph library called JGraphT (http://www.jgrapht.org/, jar files and a usage demo are available on the course webpage). To simplify things, the method has been defined to use only graphs whose vertices are represented by String types.

Hand in all of your source code via the webhandin program. Provide a short write-up of how your program works—what algorithm(s) did you use? What is the running time of your algorithm?
**Program 2: Huffman Coding**

For this program you will implement Huffman Coding. You may choose to implement it in any language using any framework you like as long as it does not make the assignment trivial. The program will take, as input, a file name. It will open the file (a plaintext ASCII file) and process it, producing a prefix-free, variable length binary encoding that minimizes the average codeword length.

Your program should output: the encoding, the average codeword length, the total number of bits in the original file (simply the total number of characters multiplied by 8), the total number of bits in the encoded file, and the compression ratio. Your output should resemble something like:

<table>
<thead>
<tr>
<th>Character</th>
<th>Codeword</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>011110111100</td>
<td>0.0502%</td>
</tr>
<tr>
<td>,</td>
<td>101011111000</td>
<td>0.0335%</td>
</tr>
<tr>
<td>-</td>
<td>1101010</td>
<td>1.0038%</td>
</tr>
<tr>
<td>.</td>
<td>011110110000</td>
<td>0.2008%</td>
</tr>
<tr>
<td>2</td>
<td>110101111000010</td>
<td>0.0167%</td>
</tr>
</tbody>
</table>