Money Change Problem (Example 1.24)

Currency System with following coins:
- Dollars, Quarters, Dimes, Nickels, Pennies

Want to give change of "n" value in cents such that total number of
coins is minimized. Give greedy algorithm to solve problem.

```
1. Defn: Change(n) ==
2.  "int quarters = 0;
3.  "int dimes = 0;
4.  "int nickels = 0;
5.  "int pennies = 0;
6.  "while (n >= 25)
7.      "quarters++;
8.      "n -= 25;
9.  "while (n >= 10)
10.     "dimes++;
11.     "n -= 10;
12.  "while (n >= 5)
13.     "nickels++;
14.     "n -= 5;
15.  "while (n >= 1)
16.     "pennies++;
17.     "n -= 1;
18. "return quarters + dimes + nickels + pennies;
```

Example:
- Change(97)
- 0 quarters
- 1 dime
- 1 nickel
- 2 pennies
- Total: 3 coins
Problem 8.4

Give a counterexample to show greedy algorithm from 8.3 is not always correct if coins are 1, 5, 7, 11 cents.

Using Greedy          Example, \( n = 14 \)
\[
\begin{align*}
\text{n} &= 14 \geq 11 \checkmark \quad \text{Coin Count} = 1 \\
\text{n} &= 14 - 11 = 3 \\
\text{n} &= 3 \geq 7 \times \\
\text{n} &= 3 \geq 5 \times \\
\text{n} &= 3 \geq 1 \checkmark \quad \text{Coin Count} = 2 \\
\text{3-1 = 2} \\
\text{n} &= 2 \geq 1 \checkmark \quad \text{Coin Count} = 3 \\
\text{2-1 = 1} \\
\text{n} &= 12 \checkmark \quad \text{Coin Count} = 4 \\
\text{n} &= 0 \quad \text{Done} \quad \underline{4 \text{ Coins}}
\end{align*}
\]

Optimal
\[
7 + 7 = 14 \checkmark \quad \underline{2 \text{ Coins}}
\]

When \( n = 14 \ldots \)

Greedy = 4 coins
Optimal = 2 coins
Problem 8.8

Let \( G = (V, E) \) be an undirected graph. A clique \( C \) in \( G \) is a subgraph in \( G \) that is a complete graph by itself. A clique \( C \) is maximum if there is no other clique \( C' \) in \( G \) such that the size of \( C' \) is greater than the size of \( C \). Consider the following method to find max clique in \( G \).

Let \( C = G \). Repeat following until \( C \) is a clique. Delete from \( C \) a vertex that is not connected to every other vertex in \( C \).

Show that this greedy approach does not always result in max clique.

Show by example:

Following algorithm:

1. 4 not connected to 2 \( \rightarrow \) remove 4
2. 2 not connected to 5 \( \rightarrow \) remove 2
3. 5 not connected to 3 \( \rightarrow \) remove 5

Clique (All connected) size = 3

Best Result:

Clique (All connected) size = 4

Greedy is not always max clique
8.10) Give a greedy algorithm for the order in which these arrays should be merged so that the overall # of comparisons is minimized.

Consider the sizes $n_i$'s corresponding to the arrays $A_j$'s.

1. Sort $n_1, n_2, \ldots, n_j$ in non-decreasing order.

2. Sort the first two arrays that correspond to the smallest two sizes, say $n_i, n_j$ ($A_i, A_j$) as $A_i$.

3. Find the position of the size $n_i$ of $A_i$ in the sorted list of sizes.

4. Repeat steps (2-3) until sorted.

Note that in the Huffman Algorithm, the "set of characters" is the set of arrays and the "frequencies" are the respective sizes.

// Insert the arrays into a min heap $H$ according to their sizes.
// Huffman tree $(T, T)$ for $\{n\}$ set of arrays

$V \leftarrow \{d\}; T = \emptyset$

For $j = 1$ to $n-1$

- $a \leftarrow \text{DeleteMin}(H)$
- $a' \leftarrow \text{DeleteMin}(H)$
- $\text{size}(v) = \text{size}(a) + \text{size}(a')$ // $v$ is a new node
  - Insert $(H, v)$
- $V = V \cup \{v\}$ // add $v$ to $V$
- $T = T \cup S(v, a), (v, a')$ // make the arrays $a$ & $a'$ children of $v$ in $T$
8.10. Analyze the time complexity of the algorithm in (8.10).

Repealing the merge of two arrays induces a binary tree in which each node is a merge. The contribution of any leaf of the tree, to the total cost of the final merge is the weight of that leaf times its depth. (bc each node is a merge, and the values in the leaf take part of the merges in the path from the leaf to the root.)

This is similar to the total lengths of a Huffman encoding: the sum of the products of the frequency of a symbol with the depth of the leaf corresponding to that symbol.

The time complexity is: $O(n \log m)$ where $m$ is the # of arrays to be merged.
Apply Dijkstra on the following graph. Assume that vertex 1 is the start vertex.

\[ X = 8, 1, 3, 2, 5, 4, 8 \]

\[ X = 8, 1, 3, 2, 5, 8 \]

\[ X = 8, 1, 3, 2, 5, 8 \]

\[ X = 8, 1, 3, 2, 5, 8 \]
Problem 8.23
Show the result of applying Kruskal's algorithm to find minimum cost spanning tree for undirected graph.

Go until added n-1 edges.
\( n = 6 \): 5 edges

<table>
<thead>
<tr>
<th>Edge</th>
<th>Distance</th>
<th>Taken?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>1</td>
<td>✓</td>
</tr>
<tr>
<td>(1,4)</td>
<td>2</td>
<td>x</td>
</tr>
<tr>
<td>(2,4)</td>
<td>2</td>
<td>✓</td>
</tr>
<tr>
<td>(3,6)</td>
<td>3</td>
<td>✓</td>
</tr>
<tr>
<td>(5,6)</td>
<td>3</td>
<td>x</td>
</tr>
<tr>
<td>(3,5)</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>(4,6)</td>
<td>6</td>
<td>✓</td>
</tr>
<tr>
<td>(4,3)</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>(4,5)</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Result

Cost: 15
Problem 8.24:

Show result of applying Prim algorithm to find min cost spanning tree for undirected graph.

Start vertex 1

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Choose 2

Choose 4 (Defaut form 1)

Choose 5

Done

Choose 3 (Default choice) from 1

Choose 6

Final Result

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

Cost = 15
8:30) G directed weighted graph $G$:
No two edges have the same weight.
Let $T$ be the shortest path tree.
Let $G'$ be the undirected graph obtained from $G$. Let $T'$ be the minimum spanning tree.
Prove or disprove $T = T'$.

- Consider the following graph $G$:

- Finding the shortest path tree with vertex 0 as source gives:

- But the minimum spanning tree is:

\[ X = 30, 13 \]

\[ X = 30, 1, 2, 8 \]

\[ \therefore T \neq T' \]
Problem 8.31

Use Huffman algorithm to find an optimal code for characters a, b, c, d, e, f whose freq are 7, 5, 3, 2, 12, 9.

Merge a:7/c:3

Merge a:7/b:5

Merge a:7/f:9

Merge a:7/b:5

Merge a:7/16

Merge a:7/e:12

Merge a:7/f:9

Code:

- a: 00
- b: 101
- c: 1001
- d: 11
- e: 01
- f: 0