UNIVERSITY OF ALABAMA
COLLEGE OF ENGINEERING

Homework #1

Student name: *Andrew Hankins*

Course: *Artificial Intelligence (CS 565)* – Professor: *Dr. Monica Anderson Herzog*
Due date: *February 8th, 2023*

1. Updated Python Files

achankins.py

```python
from src.strategies import Strategy
from src.piece import Piece
from src.compare_all_moves_strategy import CompareAllMoves

import src.weight
import src.hash

class player1_achankins(CompareAllMoves):

    def evaluate_board(self, myboard, colour):
        board_stats = self.assess_board(colour, myboard)
        # Attempt to normalize the features between a value of 0...1 and weight them
        board_value = 0.75 * (board_stats['sum_distances'] / 163.0) + \
                      -0.75 * (board_stats['number_of_singles'] / 7.0) + \
                      -0.75 * (board_stats['number_occupied_spaces'] / 7.0) + \
                      -0.25 * (board_stats['opponents_taken_pieces'] / 1.0) + \
                       0.9 * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                       0.9 * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                       1.0 * (board_stats['pieces_on_board'] / 15.0) + \
                      -1.0 * (board_stats['sum_distances_opponent'] / 163.0)
        return board_value

class player2_achankins(CompareAllMoves):

    def evaluate_board(self, myboard, colour):
        board_stats = self.assess_board(colour, myboard)
        weight_list = src.weight.weight
        # Attempt to normalize the features between a value of 0...1 and weight them
        board_value = float(weight_list[0]) * (board_stats['sum_distances'] / 163.0) + \
                      float(weight_list[1]) * (board_stats['number_of_singles'] / 7.0) + \
                      float(weight_list[2]) * (board_stats['number_occupied_spaces'] / 7.0) + \
                      float(weight_list[3]) * (board_stats['opponents_taken_pieces'] / 1.0) + \
                      float(weight_list[4]) * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                      float(weight_list[5]) * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                      float(weight_list[6]) * (board_stats['pieces_on_board'] / 15.0) + \
                      float(weight_list[7]) * (board_stats['sum_distances_opponent'] / 163.0) + \
                      float(weight_list[8]) * (board_stats['num_locations_with_two_or_three_pieces'] / 7.0)

        return board_value
```

compare_all_moves_strategy.py

```
from src.strategies import Strategy
from src.piece import Piece


class CompareAllMoves(Strategy):

    @staticmethod
    def get_difficulty():
        return "Hard"

    # Function that generates the features to be used when calculating the best
    # possible move.
    def assess_board(self, colour, myboard):
        # Get the current location of the pieces on the board
        pieces = myboard.get_pieces(colour)
        # Get the number of pieces on the board
        pieces_on_board = len(pieces)
        # Initialize the features that will be returned by the function
        sum_distances = 0
        number_of_singles = 0
        number_occupied_spaces = 0
        sum_single_distance_away_from_home = 0
        sum_distances_to_endzone = 0
        num_locations_with_two_or_three_pieces = 0
        # Calculate the sum of the pieces distance to home and the sum of the
        # pieces distance to the endzone (last section of board)
        for piece in pieces:
            sum_distances = sum_distances + piece.spaces_to_home()
            if piece.spaces_to_home() > 6:
                sum_distances_to_endzone += piece.spaces_to_home() - 6
        # Get the number of single pieces, the sum of the single pieces distance
        # to home, and the number of occupied spaces.
        for location in range(1, 25):
            pieces = myboard.pieces_at(location)
            if len(pieces) != 0 and pieces[0].colour == colour:
                if len(pieces) == 1:
                    number_of_singles = number_of_singles + 1
                    sum_single_distance_away_from_home += 25 - pieces[0].spaces_to_home()
                elif len(pieces) > 1: # Not counting single spaces
                    number_occupied_spaces = number_occupied_spaces + 1
                if len(pieces) > 1 and len(pieces) <= 3:
                    num_locations_with_two_or_three_pieces += 1
        # Get the number of piece's we have taken from the opponent
        opponents_taken_pieces = len(myboard.get_taken_pieces(colour.other()))
        # Get the number of opponent's pieces on the board
        opponent_pieces = myboard.get_pieces(colour.other())
        # Get the sum of the opponents pieces to their home
        sum_distances_opponent = 0
        for piece in opponent_pieces:
            sum_distances_opponent = sum_distances_opponent + piece.spaces_to_home()

        """
        # Calculate the probability that our single pieces can be taken
        probability_piece_can_be_taken = 0
        for location in range(1, 25):
            pieces1 = myboard.pieces_at(location)
            if len(pieces1) == 1 and pieces1[0].colour == colour:
                for idx in range (25, location, -1):
                    if pieces1[0] != colour:
                        # Calculate the opponent is from our location
                        distance_to_single_piece = idx - location
                        if distance_to_single_piece == 12: # (1/36 chance)
                            probability_piece_can_be_taken += 1/36
                        elif distance_to_single_piece == 11: # (2/36 chance)
                            probability_piece_can_be_taken += 2/36
                        elif distance_to_single_piece == 10: # (3/36 chance)
                            probability_piece_can_be_taken += 3/36
                        elif distance_to_single_piece == 9: # (4/36 chance)
                            probability_piece_can_be_taken += 4/36
                        elif distance_to_single_piece == 8: # (5/36 chance)
                            probability_piece_can_be_taken += 5/36
                        elif distance_to_single_piece == 7: # (6/36 chance)
                            probability_piece_can_be_taken += 6/36
                        elif distance_to_single_piece == 6: # (5/36 chance)
                            probability_piece_can_be_taken += 5/36
                        elif distance_to_single_piece == 5: # (4/36 chance)
                            probability_piece_can_be_taken += 4/36
                        elif distance_to_single_piece == 4: # (3/36 chance)
                            probability_piece_can_be_taken += 3/36
                        elif distance_to_single_piece == 3: # (2/36 chance)
                            probability_piece_can_be_taken += 2/36
                        elif distance_to_single_piece == 2: # (1/36 chance)
                            probability_piece_can_be_taken += 1/36
        """

        # New feature calculation (Pieces in best quadrant)
        """
        num_pieces_in_best_locations = 0
        for location in range(1, 25):
            pieces = myboard.pieces_at(location)
            if len(pieces) > 1 and len(pieces) <=3 and ((location == 5) or (location == 20)):
                num_pieces_in_best_locations += 1
        """

        return {
```

compare_all_moves_strategy.py

```
            'number_occupied_spaces': number_occupied_spaces,
            'opponents_taken_pieces': opponents_taken_pieces,
            'sum_distances': sum_distances,
            'sum_distances_opponent': sum_distances_opponent,
            'number_of_singles': number_of_singles,
            'sum_single_distance_away_from_home': sum_single_distance_away_from_home,
            'pieces_on_board': pieces_on_board,
            'sum_distances_to_endzone': sum_distances_to_endzone,
            'num_locations_with_two_or_three_pieces': num_locations_with_two_or_three_pieces
        }

    # Function that will start the process to determine the best move, then
    # move the piece
    def move(self, board, colour, dice_roll, make_move, opponents_activity):

        # Determine the best move available
        result = self.move_recursively(board, colour, dice_roll)
        # If the roll is a double then the length will be 4
        not_a_double = len(dice_roll) == 2
        # If the roll is not a double then also check the dice in the reverse
        # order to ensure we currently have chosen the best possible move
        if not_a_double:
            new_dice_roll = dice_roll.copy()
            new_dice_roll.reverse()
            result_swapped = self.move_recursively(board, colour,
                                                   dice_rolls=new_dice_roll)
            if result_swapped['best_value'] < result['best_value'] and \
                    len(result_swapped['best_moves']) >= len(result['best_moves']):
                result = result_swapped

        # Make the best move(s)
        if len(result['best_moves']) != 0:
            for move in result['best_moves']:
                make_move(move['piece_at'], move['die_roll'])

    # Function that will recursively check for the best move
    def move_recursively(self, board, colour, dice_rolls):
        best_board_value = float('inf')
        best_pieces_to_move = []

        # Get the players current pieces
        pieces_to_try = [x.location for x in board.get_pieces(colour)]
        pieces_to_try = list(set(pieces_to_try))

        # Get one piece from each location to test
        valid_pieces = []
        for piece_location in pieces_to_try:
            valid_pieces.append(board.get_piece_at(piece_location))
        valid_pieces.sort(key=Piece.spaces_to_home, reverse=True)

        # Get the first dice roll
        dice_rolls_left = dice_rolls.copy()
        die_roll = dice_rolls_left.pop(0)

        # Iterate through each piece and test possible moves
        for piece in valid_pieces:
            if board.is_move_possible(piece, die_roll):
                board_copy = board.create_copy()
                new_piece = board_copy.get_piece_at(piece.location)
                board_copy.move_piece(new_piece, die_roll)
                if len(dice_rolls_left) > 0:
                    result = self.move_recursively(board_copy, colour, dice_rolls_left)
                    if len(result['best_moves']) == 0:
                        # we have done the best we can do
                        board_value = self.evaluate_board(board_copy, colour)
                        if board_value < best_board_value and len(best_pieces_to_move) < 2:
                            best_board_value = board_value
                            best_pieces_to_move = [{'die_roll': die_roll, 'piece_at': piece.location}]
                    elif result['best_value'] < best_board_value:
                        new_best_moves_length = len(result['best_moves']) + 1
                        if new_best_moves_length >= len(best_pieces_to_move):
                            best_board_value = result['best_value']
                            move = {'die_roll': die_roll, 'piece_at': piece.location}
                            best_pieces_to_move = [move] + result['best_moves']
                else:
                    board_value = self.evaluate_board(board_copy, colour)
                    if board_value < best_board_value and len(best_pieces_to_move) < 2:
                        best_board_value = board_value
                        best_pieces_to_move = [{'die_roll': die_roll, 'piece_at': piece.location}]

        return {'best_value': best_board_value,
                'best_moves': best_pieces_to_move}


class CompareAllMovesSimple(CompareAllMoves):

    def evaluate_board(self, myboard, colour):
        board_stats = self.assess_board(colour, myboard)

        board_value = board_stats['sum_distances'] + 2 * board_stats['number_of_singles'] - \
                      board_stats['number_occupied_spaces'] - board_stats['opponents_taken_pieces']
        return board_value


class CompareAllMovesWeightingDistance(CompareAllMoves):

    def evaluate_board(self, myboard, colour):
        board_stats = self.assess_board(colour, myboard)
        board_value = board_stats['sum_distances'] - float(board_stats['sum_distances_opponent'])/3 + \
                      2 * board_stats['number_of_singles'] - \
```

2. Explanation of Novel Feature

3. Comparison of 5 Best Weighting Functions

The following section will discuss the five best weighting functions that I tested throughout my searching process.

3.1. Best Weighting Function.

The first weighting function that I found

```
class player1_achankins(CompareAllMoves):

    # Function that will evaluate the board
    def evaluate_board(self, myboard, colour):
    board_stats = self.assess_board(colour, myboard)

    # Attempt to normalize the features between a value of 0...1 and weight them
    board_value = 0.75 * (board_stats['sum_distances'] / 163.0) + \
                          -0.75 * (board_stats['number_of_singles'] / 7.0) + \
                          -0.75 * (board_stats['number_occupied_spaces'] / 7.0) + \
                          -0.25 * (board_stats['opponents_taken_pieces'] / 1.0) + \
                           0.9 * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                           0.9 * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                           1.0 * (board_stats['pieces_on_board'] / 15.0) + \
                          -1.0 * (board_stats['sum_distances_opponent'] / 163.0)
    return board_value
```

Opponent	Run 1	Run 2	Run 3	Avg. Win Rate	Std. Dev.
CAMWD	6	10	15	100%	1
MFBS	2	3	4	100%	1

Table 1: Weighting algorithm 1

3.2. Second Best Weighting Function.

The second weighting function that I found

```
class player1_achankins(CompareAllMoves):

    # Function that will evaluate the board
    def evaluate_board(self, myboard, colour):
    board_stats = self.assess_board(colour, myboard)

    # Attempt to normalize the features between a value of 0...1 and weight them
    board_value = 0.75 * (board_stats['sum_distances'] / 163.0) + \
                          -0.75 * (board_stats['number_of_singles'] / 7.0) + \
                          -0.75 * (board_stats['number_occupied_spaces'] / 7.0) + \
                          -0.25 * (board_stats['opponents_taken_pieces'] / 1.0) + \
                           0.9 * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                           0.9 * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                           1.0 * (board_stats['pieces_on_board'] / 15.0) + \
                          -1.0 * (board_stats['sum_distances_opponent'] / 163.0)
    return board_value
```

Opponent	Run 1	Run 2	Run 3	Avg. Win Rate	Std Dev.
CAMWD	6	10	15	100%	1
MFBS	2	3	4	100%	1

Table 2: Weighting algorithm 2

3.3. Third Best Weighting Function.

The third weighting function that I found

```
class player1_achankins(CompareAllMoves):

    # Function that will evaluate the board
    def evaluate_board(self, myboard, colour):
    board_stats = self.assess_board(colour, myboard)

    # Attempt to normalize the features between a value of 0...1 and weight them
    board_value = 0.75 * (board_stats['sum_distances'] / 163.0) + \
                          -0.75 * (board_stats['number_of_singles'] / 7.0) + \
                          -0.75 * (board_stats['number_occupied_spaces'] / 7.0) + \
                          -0.25 * (board_stats['opponents_taken_pieces'] / 1.0) + \
                           0.9 * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                           0.9 * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                           1.0 * (board_stats['pieces_on_board'] / 15.0) + \
                          -1.0 * (board_stats['sum_distances_opponent'] / 163.0)
    return board_value
```

Opponent	Run 1	Run 2	Run 3	Avg. Win Rate	Std Dev.
CAMWD	6	10	15	100%	1
MFBS	2	3	4	100%	1

Table 3: Weighting algorithm 3

3.4. Fourth Best Weighting Function.

The fourth weighting function that I found

```
class player1_achankins(CompareAllMoves):

    # Function that will evaluate the board
    def evaluate_board(self, myboard, colour):
    board_stats = self.assess_board(colour, myboard)

    # Attempt to normalize the features between a value of 0...1 and weight them
    board_value = 0.75 * (board_stats['sum_distances'] / 163.0) + \
                          -0.75 * (board_stats['number_of_singles'] / 7.0) + \
                          -0.75 * (board_stats['number_occupied_spaces'] / 7.0) + \
                          -0.25 * (board_stats['opponents_taken_pieces'] / 1.0) + \
                           0.9 * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                           0.9 * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                           1.0 * (board_stats['pieces_on_board'] / 15.0) + \
                          -1.0 * (board_stats['sum_distances_opponent'] / 163.0)
    return board_value
```

Opponent	Run 1	Run 2	Run 3	Avg. Win Rate	Std Dev.
CAMWD	6	10	15	100%	1
MFBS	2	3	4	100%	1

Table 4: Weighting algorithm 4

3.5. Fifth Best Weighting Function.

The fifth weighting function that I found

```
class player1_achankins(CompareAllMoves):

    # Function that will evaluate the board
    def evaluate_board(self, myboard, colour):
    board_stats = self.assess_board(colour, myboard)

    # Attempt to normalize the features between a value of 0...1 and weight them
    board_value = 0.75 * (board_stats['sum_distances'] / 163.0) + \
                          -0.75 * (board_stats['number_of_singles'] / 7.0) + \
```

```
                        −0.75 * (board_stats['number_occupied_spaces'] / 7.0) + \
                        −0.25 * (board_stats['opponents_taken_pieces'] / 1.0) + \
                         0.9 * (board_stats['sum_distances_to_endzone'] / 75.0) + \
                         0.9 * (board_stats['sum_single_distance_away_from_home'] / 100.0) + \
                         1.0 * (board_stats['pieces_on_board'] / 15.0) + \
                        −1.0 * (board_stats['sum_distances_opponent'] / 163.0)
    return board_value
```

Opponent	Run 1	Run 2	Run 3	Avg. Win Rate	Std. Dev.
CAMWD	6	10	15	100%	1
MFBS	2	3	4	100%	1

Table 5: Weighting algorithm 5

Player Comparisons

The following section compares the performance of both player1_achankins and player2_achankins against the MoveFurthestBackStrategy and CompareAllMovesWeightingDistance players. This test was done by running three sets of 200 games per player per opponent. After each run, the winning percentage of the player was recorded. Once all three runs had been completed, the average win percentage and standard deviation was calculated. This testing system was designed to accurately assess the newly created players by using a sufficient amount of games multiple times in order to ensure the result was correct.

Player	Run 1	Run 2	Run 3	Avg. Win Rate	Std. Dev.
player1_achankins	96.0%	94.0%	94.5%	94.83%	0.85
player2_achankins	2	3	4	100%	1

Table 6: Comparison against MoveFurthestBackStrategy

Player	Run 1	Run 2	Run 3	Avg. Win Rate	Std. Dev.
Player1_achankins	61.5%	61.5%	65.5%	62.83%	1.89
player2_achankins	2	3	4	100%	1

Table 7: Comparison against CompareAllMovesWeightingDistance

Game Tree

The minimax algorithm is a strategy designed to select the optimal move in an adversarial game by assuming the opponent always selects the move that will minimize your score. By looking ahead we are able to see the worst case scenario from each roll and select the best possible course of action. For example from the below minimax tree we can determine that going to state 10 will give us the best possible outcome.

```
Initial State:
Roll: [5,2]
Resulting State | Move1    | Move2    | Current Utility | Avg Minimum Utility
================================================================================
State:  1       | [2, 7]   | [5, 7]   | 107.0           | 108.9166675
State:  2       | [2, 7]   | [7, 9]   | 107.0           | 108.9166675
State:  3       | [2, 7]   | [12, 14] | 109.0           | 110.9166675
State:  4       | [2, 7]   | [17, 19] | 110.0           | 111.9166675
State:  5       | [2, 7]   | [19, 21] | 109.0           | 110.9166675
State:  6       | [5, 10]  | [2, 4]   | 112.0           | 113.9166675
State:  7       | [5, 10]  | [5, 7]   | 112.0           | 113.9166675
State:  8       | [5, 10]  | [10, 12] | 110.0           | 111.9166675
State:  9       | [5, 10]  | [12, 14] | 114.0           | 115.9166675
State: 10       | [5, 10]  | [17, 19] | 115.0           | 116.9166675
State: 11       | [5, 10]  | [19, 21] | 114.0           | 115.9166675
State: 12       | [12, 17] | [2, 4]   | 107.0           | 108.9166675
State: 13       | [12, 17] | [5, 7]   | 112.0           | 113.9166675
State: 14       | [12, 17] | [12, 14] | 109.0           | 110.9166675
State: 15       | [12, 17] | [17, 19] | 107.0           | 108.9166675
State: 16       | [12, 17] | [19, 21] | 109.0           | 110.9166675
State: 17       | [19, 24] | [2, 4]   | 109.0           | 110.9166675
State: 18       | [19, 24] | [5, 7]   | 114.0           | 115.9166675
State: 19       | [19, 24] | [12, 14] | 111.0           | 112.9166675
State: 20       | [19, 24] | [17, 19] | 112.0           | 113.9166675
State: 21       | [19, 24] | [19, 21] | 111.0           | 112.9166675

Previous State: 1
Roll: [1,6]
State 1: [3, 2] [15, 9] 109.33
State 2: [3, 2] [22, 16] 109.33
State 3: [6, 5] [15, 9] 109.33
State 4: [6, 5] [22, 16] 109.33
State 5: [15, 14] [14, 8] 109.33
State 6: [15, 14] [22, 16] 109.33
State 7: [22, 21] [15, 9] 109.33
State 8: [22, 21] [21, 15] 109.33
State 9: [22, 21] [22, 16] 109.33
Min Utility: 109.33333

Previous State: 1
Roll: [3,5]
State 1: [6, 3] [6, 1] 109.67
State 2: [6, 3] [13, 8] 109.67
State 3: [6, 3] [15, 10] 109.67
State 4: [13, 10] [6, 1] 109.67
State 5: [13, 10] [10, 5] 109.67
State 6: [13, 10] [13, 8] 109.67
State 7: [13, 10] [15, 10] 109.67
Min Utility: 109.66667

Previous State: 1
Roll: [2,3]
State 1: [3, 1] [6, 3] 108.67
State 2: [3, 1] [13, 10] 108.67
State 3: [6, 4] [6, 3] 108.67
State 4: [6, 4] [13, 10] 108.67
State 5: [13, 11] [6, 3] 108.67
State 6: [13, 11] [11, 8] 108.67
State 7: [13, 11] [13, 10] 108.67
State 8: [15, 13] [6, 3] 108.67
State 9: [15, 13] [13, 10] 108.67
State 10: [22, 20] [6, 3] 108.67
State 11: [22, 20] [13, 10] 108.67
Min Utility: 108.66667

Previous State: 1
Roll: [1,2]
State 1: [3, 2] [3, 1] 108.0
State 2: [3, 2] [6, 4] 108.0
State 3: [3, 2] [13, 11] 108.0
State 4: [3, 2] [15, 13] 108.0
State 5: [3, 2] [22, 20] 108.0
State 6: [6, 5] [3, 1] 108.0
State 7: [6, 5] [5, 3] 108.0
State 8: [6, 5] [6, 4] 108.0
State 9: [6, 5] [13, 11] 108.0
State 10: [6, 5] [15, 13] 108.0
State 11: [6, 5] [22, 20] 108.0
State 12: [15, 14] [3, 1] 108.0
State 13: [15, 14] [6, 4] 108.0
State 14: [15, 14] [13, 11] 108.0
State 15: [15, 14] [22, 20] 108.0
State 16: [22, 21] [3, 1] 108.0
State 17: [22, 21] [6, 4] 108.0
State 18: [22, 21] [13, 11] 108.0
```

```
State 19: [22, 21] [15, 13] 108.0
State 20: [22, 21] [22, 20] 108.0
State 21: [13, 11] [11, 10] 108.0
Min Utility: 108.0

Previous State: 2
Roll: [1,6]
State 1: [3, 2] [13, 7] 109.33
State 2: [3, 2] [15, 9] 109.33
State 3: [3, 2] [22, 16] 109.33
State 4: [15, 14] [13, 7] 109.33
State 5: [15, 14] [14, 8] 109.33
State 6: [15, 14] [22, 16] 109.33
State 7: [22, 21] [13, 7] 109.33
State 8: [22, 21] [15, 9] 109.33
State 9: [22, 21] [21, 15] 109.33
State 10: [22, 21] [22, 16] 109.33
State 11: [13, 7] [7, 6] 109.33
Min Utility: 109.33333

Previous State: 2
Roll: [3,5]
State 1: [6, 3] [6, 1] 109.67
State 2: [6, 3] [13, 8] 109.67
State 3: [6, 3] [15, 10] 109.67
State 4: [13, 10] [6, 1] 109.67
State 5: [13, 10] [13, 8] 109.67
State 6: [13, 10] [15, 10] 109.67
State 7: [15, 10] [10, 7] 109.67
Min Utility: 109.66667

Previous State: 2
Roll: [2,3]
State 1: [3, 1] [6, 3] 108.67
State 2: [3, 1] [13, 10] 108.67
State 3: [6, 4] [6, 3] 108.67
State 4: [6, 4] [13, 10] 108.67
State 5: [13, 11] [6, 3] 108.67
State 6: [13, 11] [11, 8] 108.67
State 7: [13, 11] [13, 10] 108.67
State 8: [15, 13] [6, 3] 108.67
State 9: [15, 13] [13, 10] 108.67
State 10: [22, 20] [6, 3] 108.67
State 11: [22, 20] [13, 10] 108.67
Min Utility: 108.66667

Previous State: 2
Roll: [1,2]
State 1: [3, 2] [3, 1] 108.0
State 2: [3, 2] [6, 4] 108.0
State 3: [3, 2] [13, 11] 108.0
State 4: [3, 2] [15, 13] 108.0
State 5: [3, 2] [22, 20] 108.0
State 6: [15, 14] [3, 1] 108.0
State 7: [15, 14] [6, 4] 108.0
State 8: [15, 14] [13, 11] 108.0
State 9: [15, 14] [22, 20] 108.0
State 10: [22, 21] [3, 1] 108.0
State 11: [22, 21] [6, 4] 108.0
State 12: [22, 21] [13, 11] 108.0
State 13: [22, 21] [15, 13] 108.0
State 14: [22, 21] [22, 20] 108.0
State 15: [6, 4] [4, 3] 108.0
State 16: [13, 11] [11, 10] 108.0
Min Utility: 108.0

Previous State: 3
Roll: [1,6]
State 1: [3, 2] [13, 7] 111.33
State 2: [3, 2] [15, 9] 111.33
State 3: [3, 2] [22, 16] 111.33
State 4: [15, 14] [13, 7] 111.33
State 5: [15, 14] [14, 8] 111.33
State 6: [15, 14] [22, 16] 111.33
State 7: [22, 21] [13, 7] 111.33
State 8: [22, 21] [15, 9] 111.33
State 9: [22, 21] [21, 15] 111.33
State 10: [22, 21] [22, 16] 111.33
State 11: [13, 7] [7, 6] 111.33
Min Utility: 111.33333

Previous State: 3
Roll: [3,5]
State 1: [6, 3] [6, 1] 111.67
State 2: [6, 3] [13, 8] 111.67
State 3: [6, 3] [15, 10] 111.67
State 4: [13, 10] [6, 1] 111.67
State 5: [13, 10] [13, 8] 111.67
State 6: [13, 10] [15, 10] 111.67
State 7: [15, 10] [10, 7] 111.67
Min Utility: 111.66667

Previous State: 3
Roll: [2,3]
State 1: [3, 1] [6, 3] 110.67
State 2: [3, 1] [13, 10] 110.67
State 3: [6, 4] [6, 3] 110.67
State 4: [6, 4] [13, 10] 110.67
State 5: [13, 11] [6, 3] 110.67
```

```
State 6: [13, 11] [11, 8] 110.67
State 7: [13, 11] [13, 10] 110.67
State 8: [15, 13] [6, 3] 110.67
State 9: [15, 13] [13, 10] 110.67
State 10: [22, 20] [6, 3] 110.67
State 11: [22, 20] [13, 10] 110.67
Min Utility: 110.66667

Previous State: 3
Roll: [1,2]
State 1: [3, 2] [3, 1] 110.0
State 2: [3, 2] [6, 4] 110.0
State 3: [3, 2] [13, 11] 110.0
State 4: [3, 2] [15, 13] 110.0
State 5: [3, 2] [22, 20] 110.0
State 6: [15, 14] [3, 1] 110.0
State 7: [15, 14] [6, 4] 110.0
State 8: [15, 14] [13, 11] 110.0
State 9: [15, 14] [22, 20] 110.0
State 10: [22, 21] [3, 1] 110.0
State 11: [22, 21] [6, 4] 110.0
State 12: [22, 21] [13, 11] 110.0
State 13: [22, 21] [15, 13] 110.0
State 14: [22, 21] [22, 20] 110.0
State 15: [6, 4] [4, 3] 110.0
State 16: [13, 11] [11, 10] 110.0
Min Utility: 110.0

Previous State: 4
Roll: [1,6]
State 1: [3, 2] [13, 7] 112.33
State 2: [3, 2] [15, 9] 112.33
State 3: [3, 2] [22, 16] 112.33
State 4: [15, 14] [13, 7] 112.33
State 5: [15, 14] [14, 8] 112.33
State 6: [15, 14] [22, 16] 112.33
State 7: [22, 21] [13, 7] 112.33
State 8: [22, 21] [15, 9] 112.33
State 9: [22, 21] [21, 15] 112.33
State 10: [22, 21] [22, 16] 112.33
State 11: [13, 7] [7, 6] 112.33
Min Utility: 112.33333

Previous State: 4
Roll: [3,5]
State 1: [6, 3] [6, 1] 112.67
State 2: [6, 3] [13, 8] 112.67
State 3: [6, 3] [15, 10] 112.67
State 4: [6, 3] [22, 17] 112.67
State 5: [13, 10] [6, 1] 112.67
State 6: [13, 10] [13, 8] 112.67
State 7: [13, 10] [15, 10] 112.67
State 8: [13, 10] [22, 17] 112.67
State 9: [15, 10] [10, 7] 112.67
State 10: [22, 17] [17, 14] 112.67
Min Utility: 112.66667

Previous State: 4
Roll: [2,3]
State 1: [3, 1] [6, 3] 111.67
State 2: [3, 1] [13, 10] 111.67
State 3: [6, 4] [6, 3] 111.67
State 4: [6, 4] [13, 10] 111.67
State 5: [13, 11] [6, 3] 111.67
State 6: [13, 11] [11, 8] 111.67
State 7: [13, 11] [13, 10] 111.67
State 8: [15, 13] [6, 3] 111.67
State 9: [15, 13] [13, 10] 111.67
State 10: [22, 20] [6, 3] 111.67
State 11: [22, 20] [13, 10] 111.67
State 12: [22, 20] [20, 17] 111.67
Min Utility: 111.66667

Previous State: 4
Roll: [1,2]
State 1: [3, 2] [3, 1] 111.0
State 2: [3, 2] [6, 4] 111.0
State 3: [3, 2] [13, 11] 111.0
State 4: [3, 2] [15, 13] 111.0
State 5: [3, 2] [22, 20] 111.0
State 6: [15, 14] [3, 1] 111.0
State 7: [15, 14] [6, 4] 111.0
State 8: [15, 14] [13, 11] 111.0
State 9: [15, 14] [22, 20] 111.0
State 10: [22, 21] [3, 1] 111.0
State 11: [22, 21] [6, 4] 111.0
State 12: [22, 21] [13, 11] 111.0
State 13: [22, 21] [15, 13] 111.0
State 14: [22, 21] [22, 20] 111.0
State 15: [6, 4] [4, 3] 111.0
State 16: [13, 11] [11, 10] 111.0
Min Utility: 111.0

Previous State: 5
Roll: [1,6]
State 1: [3, 2] [13, 7] 111.33
State 2: [3, 2] [15, 9] 111.33
State 3: [3, 2] [22, 16] 111.33
State 4: [15, 14] [13, 7] 111.33
```

```
State 5: [15, 14] [14, 8] 111.33
State 6: [15, 14] [22, 16] 111.33
State 7: [22, 21] [13, 7] 111.33
State 8: [22, 21] [15, 9] 111.33
State 9: [22, 21] [21, 15] 111.33
State 10: [22, 21] [22, 16] 111.33
State 11: [13, 7] [7, 6] 111.33
Min Utility: 111.33333

Previous State: 5
Roll: [3,5]
State 1: [6, 3] [6, 1] 111.67
State 2: [6, 3] [13, 8] 111.67
State 3: [6, 3] [15, 10] 111.67
State 4: [13, 10] [6, 1] 111.67
State 5: [13, 10] [13, 8] 111.67
State 6: [13, 10] [15, 10] 111.67
State 7: [15, 10] [10, 7] 111.67
Min Utility: 111.66667

Previous State: 5
Roll: [2,3]
State 1: [3, 1] [6, 3] 110.67
State 2: [3, 1] [13, 10] 110.67
State 3: [6, 4] [6, 3] 110.67
State 4: [6, 4] [13, 10] 110.67
State 5: [13, 11] [6, 3] 110.67
State 6: [13, 11] [11, 8] 110.67
State 7: [13, 11] [13, 10] 110.67
State 8: [15, 13] [6, 3] 110.67
State 9: [15, 13] [13, 10] 110.67
State 10: [22, 20] [6, 3] 110.67
State 11: [22, 20] [13, 10] 110.67
Min Utility: 110.66667

Previous State: 5
Roll: [1,2]
State 1: [3, 2] [3, 1] 110.0
State 2: [3, 2] [6, 4] 110.0
State 3: [3, 2] [13, 11] 110.0
State 4: [3, 2] [15, 13] 110.0
State 5: [3, 2] [22, 20] 110.0
State 6: [15, 14] [3, 1] 110.0
State 7: [15, 14] [6, 4] 110.0
State 8: [15, 14] [13, 11] 110.0
State 9: [15, 14] [22, 20] 110.0
State 10: [22, 21] [3, 1] 110.0
State 11: [22, 21] [6, 4] 110.0
State 12: [22, 21] [13, 11] 110.0
State 13: [22, 21] [15, 13] 110.0
State 14: [22, 21] [22, 20] 110.0
State 15: [6, 4] [4, 3] 110.0
State 16: [13, 11] [11, 10] 110.0
Min Utility: 110.0

Previous State: 6
Roll: [1,6]
State 1: [3, 2] [13, 7] 114.33
State 2: [3, 2] [15, 9] 114.33
State 3: [3, 2] [22, 16] 114.33
State 4: [6, 5] [13, 7] 114.33
State 5: [6, 5] [15, 9] 114.33
State 6: [6, 5] [22, 16] 114.33
State 7: [15, 14] [13, 7] 114.33
State 8: [15, 14] [14, 8] 114.33
State 9: [15, 14] [22, 16] 114.33
State 10: [22, 21] [13, 7] 114.33
State 11: [22, 21] [15, 9] 114.33
State 12: [22, 21] [21, 15] 114.33
State 13: [22, 21] [22, 16] 114.33
State 14: [13, 7] [7, 6] 114.33
Min Utility: 114.33333

Previous State: 6
Roll: [3,5]
State 1: [6, 3] [6, 1] 114.67
State 2: [6, 3] [13, 8] 114.67
State 3: [6, 3] [15, 10] 114.67
State 4: [13, 10] [6, 1] 114.67
State 5: [13, 10] [10, 5] 114.67
State 6: [13, 10] [13, 8] 114.67
State 7: [13, 10] [15, 10] 114.67
State 8: [15, 10] [10, 7] 114.67
Min Utility: 114.66667

Previous State: 6
Roll: [2,3]
State 1: [3, 1] [6, 3] 113.67
State 2: [3, 1] [13, 10] 113.67
State 3: [6, 4] [6, 3] 113.67
State 4: [6, 4] [13, 10] 113.67
State 5: [13, 11] [6, 3] 113.67
State 6: [13, 11] [11, 8] 113.67
State 7: [13, 11] [13, 10] 113.67
State 8: [15, 13] [6, 3] 113.67
State 9: [15, 13] [13, 10] 113.67
State 10: [22, 20] [6, 3] 113.67
State 11: [22, 20] [13, 10] 113.67
Min Utility: 113.66667
```

```
Previous State: 6
Roll: [1,2]
State 1: [3, 2] [3, 1] 113.0
State 2: [3, 2] [6, 4] 113.0
State 3: [3, 2] [13, 11] 113.0
State 4: [3, 2] [15, 13] 113.0
State 5: [3, 2] [22, 20] 113.0
State 6: [6, 5] [3, 1] 113.0
State 7: [6, 5] [5, 3] 113.0
State 8: [6, 5] [6, 4] 113.0
State 9: [6, 5] [13, 11] 113.0
State 10: [6, 5] [15, 13] 113.0
State 11: [6, 5] [22, 20] 113.0
State 12: [15, 14] [3, 1] 113.0
State 13: [15, 14] [6, 4] 113.0
State 14: [15, 14] [13, 11] 113.0
State 15: [15, 14] [22, 20] 113.0
State 16: [22, 21] [3, 1] 113.0
State 17: [22, 21] [6, 4] 113.0
State 18: [22, 21] [13, 11] 113.0
State 19: [22, 21] [15, 13] 113.0
State 20: [22, 21] [22, 20] 113.0
State 21: [13, 11] [11, 10] 113.0
Min Utility: 113.0

Previous State: 7
Roll: [1,6]
State 1: [3, 2] [13, 7] 114.33
State 2: [3, 2] [15, 9] 114.33
State 3: [3, 2] [22, 16] 114.33
State 4: [6, 5] [13, 7] 114.33
State 5: [6, 5] [15, 9] 114.33
State 6: [6, 5] [22, 16] 114.33
State 7: [15, 14] [13, 7] 114.33
State 8: [15, 14] [14, 8] 114.33
State 9: [15, 14] [22, 16] 114.33
State 10: [22, 21] [13, 7] 114.33
State 11: [22, 21] [15, 9] 114.33
State 12: [22, 21] [21, 15] 114.33
State 13: [22, 21] [22, 16] 114.33
State 14: [13, 7] [7, 6] 114.33
Min Utility: 114.33333

Previous State: 7
Roll: [3,5]
State 1: [6, 3] [6, 1] 114.67
State 2: [6, 3] [13, 8] 114.67
State 3: [6, 3] [15, 10] 114.67
State 4: [13, 10] [6, 1] 114.67
State 5: [13, 10] [10, 5] 114.67
State 6: [13, 10] [13, 8] 114.67
State 7: [13, 10] [15, 10] 114.67
State 8: [15, 10] [10, 7] 114.67
Min Utility: 114.66667

Previous State: 7
Roll: [2,3]
State 1: [3, 1] [6, 3] 113.67
State 2: [3, 1] [13, 10] 113.67
State 3: [6, 4] [6, 3] 113.67
State 4: [6, 4] [13, 10] 113.67
State 5: [13, 11] [6, 3] 113.67
State 6: [13, 11] [11, 8] 113.67
State 7: [13, 11] [13, 10] 113.67
State 8: [15, 13] [6, 3] 113.67
State 9: [15, 13] [13, 10] 113.67
State 10: [22, 20] [6, 3] 113.67
State 11: [22, 20] [13, 10] 113.67
Min Utility: 113.66667

Previous State: 7
Roll: [1,2]
State 1: [3, 2] [3, 1] 113.0
State 2: [3, 2] [6, 4] 113.0
State 3: [3, 2] [13, 11] 113.0
State 4: [3, 2] [15, 13] 113.0
State 5: [3, 2] [22, 20] 113.0
State 6: [6, 5] [3, 1] 113.0
State 7: [6, 5] [5, 3] 113.0
State 8: [6, 5] [6, 4] 113.0
State 9: [6, 5] [13, 11] 113.0
State 10: [6, 5] [15, 13] 113.0
State 11: [6, 5] [22, 20] 113.0
State 12: [15, 14] [3, 1] 113.0
State 13: [15, 14] [6, 4] 113.0
State 14: [15, 14] [13, 11] 113.0
State 15: [15, 14] [22, 20] 113.0
State 16: [22, 21] [3, 1] 113.0
State 17: [22, 21] [6, 4] 113.0
State 18: [22, 21] [13, 11] 113.0
State 19: [22, 21] [15, 13] 113.0
State 20: [22, 21] [22, 20] 113.0
State 21: [13, 11] [11, 10] 113.0
Min Utility: 113.0

Previous State: 8
Roll: [1,6]
State 1: [3, 2] [13, 7] 112.33
```

```
State 2: [3, 2] [15, 9] 112.33
State 3: [3, 2] [22, 16] 112.33
State 4: [6, 5] [13, 7] 112.33
State 5: [6, 5] [15, 9] 112.33
State 6: [6, 5] [22, 16] 112.33
State 7: [15, 14] [13, 7] 112.33
State 8: [15, 14] [14, 8] 112.33
State 9: [15, 14] [22, 16] 112.33
State 10: [22, 21] [13, 7] 112.33
State 11: [22, 21] [15, 9] 112.33
State 12: [22, 21] [21, 15] 112.33
State 13: [22, 21] [22, 16] 112.33
State 14: [13, 7] [7, 6] 112.33
Min Utility: 112.33333

Previous State: 8
Roll: [3,5]
State 1: [6, 3] [6, 1] 112.67
State 2: [6, 3] [13, 8] 112.67
State 3: [6, 3] [15, 10] 112.67
State 4: [13, 10] [6, 1] 112.67
State 5: [13, 10] [10, 5] 112.67
State 6: [13, 10] [13, 8] 112.67
State 7: [13, 10] [15, 10] 112.67
State 8: [15, 10] [10, 7] 112.67
Min Utility: 112.66667

Previous State: 8
Roll: [2,3]
State 1: [3, 1] [6, 3] 111.67
State 2: [3, 1] [13, 10] 111.67
State 3: [6, 4] [6, 3] 111.67
State 4: [6, 4] [13, 10] 111.67
State 5: [13, 11] [6, 3] 111.67
State 6: [13, 11] [11, 8] 111.67
State 7: [13, 11] [13, 10] 111.67
State 8: [15, 13] [6, 3] 111.67
State 9: [15, 13] [13, 10] 111.67
State 10: [22, 20] [6, 3] 111.67
State 11: [22, 20] [13, 10] 111.67
Min Utility: 111.66667

Previous State: 8
Roll: [1,2]
State 1: [3, 2] [3, 1] 111.0
State 2: [3, 2] [6, 4] 111.0
State 3: [3, 2] [13, 11] 111.0
State 4: [3, 2] [15, 13] 111.0
State 5: [3, 2] [22, 20] 111.0
State 6: [6, 5] [3, 1] 111.0
State 7: [6, 5] [5, 3] 111.0
State 8: [6, 5] [6, 4] 111.0
State 9: [6, 5] [13, 11] 111.0
State 10: [6, 5] [15, 13] 111.0
State 11: [6, 5] [22, 20] 111.0
State 12: [15, 14] [3, 1] 111.0
State 13: [15, 14] [6, 4] 111.0
State 14: [15, 14] [13, 11] 111.0
State 15: [15, 14] [22, 20] 111.0
State 16: [22, 21] [3, 1] 111.0
State 17: [22, 21] [6, 4] 111.0
State 18: [22, 21] [13, 11] 111.0
State 19: [22, 21] [15, 13] 111.0
State 20: [22, 21] [22, 20] 111.0
State 21: [13, 11] [11, 10] 111.0
Min Utility: 111.0

Previous State: 9
Roll: [1,6]
State 1: [3, 2] [13, 7] 116.33
State 2: [3, 2] [15, 9] 116.33
State 3: [3, 2] [22, 16] 116.33
State 4: [6, 5] [13, 7] 116.33
State 5: [6, 5] [15, 9] 116.33
State 6: [6, 5] [22, 16] 116.33
State 7: [15, 14] [13, 7] 116.33
State 8: [15, 14] [14, 8] 116.33
State 9: [15, 14] [22, 16] 116.33
State 10: [22, 21] [13, 7] 116.33
State 11: [22, 21] [15, 9] 116.33
State 12: [22, 21] [21, 15] 116.33
State 13: [22, 21] [22, 16] 116.33
State 14: [13, 7] [7, 6] 116.33
Min Utility: 116.33333

Previous State: 9
Roll: [3,5]
State 1: [6, 3] [6, 1] 116.67
State 2: [6, 3] [13, 8] 116.67
State 3: [6, 3] [15, 10] 116.67
State 4: [13, 10] [6, 1] 116.67
State 5: [13, 10] [10, 5] 116.67
State 6: [13, 10] [13, 8] 116.67
State 7: [13, 10] [15, 10] 116.67
State 8: [15, 10] [10, 7] 116.67
Min Utility: 116.66667

Previous State: 9
Roll: [2,3]
```

```
State 1: [3, 1] [6, 3] 115.67
State 2: [3, 1] [13, 10] 115.67
State 3: [6, 4] [6, 3] 115.67
State 4: [6, 4] [13, 10] 115.67
State 5: [13, 11] [6, 3] 115.67
State 6: [13, 11] [11, 8] 115.67
State 7: [13, 11] [13, 10] 115.67
State 8: [15, 13] [6, 3] 115.67
State 9: [15, 13] [13, 10] 115.67
State 10: [22, 20] [6, 3] 115.67
State 11: [22, 20] [13, 10] 115.67
Min Utility: 115.66667

Previous State: 9
Roll: [1,2]
State 1: [3, 2] [3, 1] 115.0
State 2: [3, 2] [6, 4] 115.0
State 3: [3, 2] [13, 11] 115.0
State 4: [3, 2] [15, 13] 115.0
State 5: [3, 2] [22, 20] 115.0
State 6: [6, 5] [3, 1] 115.0
State 7: [6, 5] [5, 3] 115.0
State 8: [6, 5] [6, 4] 115.0
State 9: [6, 5] [13, 11] 115.0
State 10: [6, 5] [15, 13] 115.0
State 11: [6, 5] [22, 20] 115.0
State 12: [15, 14] [3, 1] 115.0
State 13: [15, 14] [6, 4] 115.0
State 14: [15, 14] [13, 11] 115.0
State 15: [15, 14] [22, 20] 115.0
State 16: [22, 21] [3, 1] 115.0
State 17: [22, 21] [6, 4] 115.0
State 18: [22, 21] [13, 11] 115.0
State 19: [22, 21] [15, 13] 115.0
State 20: [22, 21] [22, 20] 115.0
State 21: [13, 11] [11, 10] 115.0
Min Utility: 115.0

Previous State: 10
Roll: [1,6]
State 1: [3, 2] [13, 7] 117.33
State 2: [3, 2] [15, 9] 117.33
State 3: [3, 2] [22, 16] 117.33
State 4: [6, 5] [13, 7] 117.33
State 5: [6, 5] [15, 9] 117.33
State 6: [6, 5] [22, 16] 117.33
State 7: [15, 14] [13, 7] 117.33
State 8: [15, 14] [14, 8] 117.33
State 9: [15, 14] [22, 16] 117.33
State 10: [22, 21] [13, 7] 117.33
State 11: [22, 21] [15, 9] 117.33
State 12: [22, 21] [21, 15] 117.33
State 13: [22, 21] [22, 16] 117.33
State 14: [13, 7] [7, 6] 117.33
Min Utility: 117.33333

Previous State: 10
Roll: [3,5]
State 1: [6, 3] [6, 1] 117.67
State 2: [6, 3] [13, 8] 117.67
State 3: [6, 3] [15, 10] 117.67
State 4: [6, 3] [22, 17] 117.67
State 5: [13, 10] [6, 1] 117.67
State 6: [13, 10] [10, 5] 117.67
State 7: [13, 10] [13, 8] 117.67
State 8: [13, 10] [15, 10] 117.67
State 9: [13, 10] [22, 17] 117.67
State 10: [15, 10] [10, 7] 117.67
State 11: [22, 17] [17, 14] 117.67
Min Utility: 117.66667

Previous State: 10
Roll: [2,3]
State 1: [3, 1] [6, 3] 116.67
State 2: [3, 1] [13, 10] 116.67
State 3: [6, 4] [6, 3] 116.67
State 4: [6, 4] [13, 10] 116.67
State 5: [13, 11] [6, 3] 116.67
State 6: [13, 11] [11, 8] 116.67
State 7: [13, 11] [13, 10] 116.67
State 8: [15, 13] [6, 3] 116.67
State 9: [15, 13] [13, 10] 116.67
State 10: [22, 20] [6, 3] 116.67
State 11: [22, 20] [13, 10] 116.67
State 12: [22, 20] [20, 17] 116.67
Min Utility: 116.66667

Previous State: 10
Roll: [1,2]
State 1: [3, 2] [3, 1] 116.0
State 2: [3, 2] [6, 4] 116.0
State 3: [3, 2] [13, 11] 116.0
State 4: [3, 2] [15, 13] 116.0
State 5: [3, 2] [22, 20] 116.0
State 6: [6, 5] [3, 1] 116.0
State 7: [6, 5] [5, 3] 116.0
State 8: [6, 5] [6, 4] 116.0
State 9: [6, 5] [13, 11] 116.0
State 10: [6, 5] [15, 13] 116.0
```

```
State 11: [6, 5] [22, 20] 116.0
State 12: [15, 14] [3, 1] 116.0
State 13: [15, 14] [6, 4] 116.0
State 14: [15, 14] [13, 11] 116.0
State 15: [15, 14] [22, 20] 116.0
State 16: [22, 21] [3, 1] 116.0
State 17: [22, 21] [6, 4] 116.0
State 18: [22, 21] [13, 11] 116.0
State 19: [22, 21] [15, 13] 116.0
State 20: [22, 21] [22, 20] 116.0
State 21: [13, 11] [11, 10] 116.0
Min Utility: 116.0

Previous State: 11
Roll: [1,6]
State 1: [3, 2] [13, 7] 116.33
State 2: [3, 2] [15, 9] 116.33
State 3: [3, 2] [22, 16] 116.33
State 4: [6, 5] [13, 7] 116.33
State 5: [6, 5] [15, 9] 116.33
State 6: [6, 5] [22, 16] 116.33
State 7: [15, 14] [13, 7] 116.33
State 8: [15, 14] [14, 8] 116.33
State 9: [15, 14] [22, 16] 116.33
State 10: [22, 21] [13, 7] 116.33
State 11: [22, 21] [15, 9] 116.33
State 12: [22, 21] [21, 15] 116.33
State 13: [22, 21] [22, 16] 116.33
State 14: [13, 7] [7, 6] 116.33
Min Utility: 116.33333

Previous State: 11
Roll: [3,5]
State 1: [6, 3] [6, 1] 116.67
State 2: [6, 3] [13, 8] 116.67
State 3: [6, 3] [15, 10] 116.67
State 4: [13, 10] [6, 1] 116.67
State 5: [13, 10] [10, 5] 116.67
State 6: [13, 10] [13, 8] 116.67
State 7: [13, 10] [15, 10] 116.67
State 8: [15, 10] [10, 7] 116.67
Min Utility: 116.66667

Previous State: 11
Roll: [2,3]
State 1: [3, 1] [6, 3] 115.67
State 2: [3, 1] [13, 10] 115.67
State 3: [6, 4] [6, 3] 115.67
State 4: [6, 4] [13, 10] 115.67
State 5: [13, 11] [6, 3] 115.67
State 6: [13, 11] [11, 8] 115.67
State 7: [13, 11] [13, 10] 115.67
State 8: [15, 13] [6, 3] 115.67
State 9: [15, 13] [13, 10] 115.67
State 10: [22, 20] [6, 3] 115.67
State 11: [22, 20] [13, 10] 115.67
Min Utility: 115.66667

Previous State: 11
Roll: [1,2]
State 1: [3, 2] [3, 1] 115.0
State 2: [3, 2] [6, 4] 115.0
State 3: [3, 2] [13, 11] 115.0
State 4: [3, 2] [15, 13] 115.0
State 5: [3, 2] [22, 20] 115.0
State 6: [6, 5] [3, 1] 115.0
State 7: [6, 5] [5, 3] 115.0
State 8: [6, 5] [6, 4] 115.0
State 9: [6, 5] [13, 11] 115.0
State 10: [6, 5] [15, 13] 115.0
State 11: [6, 5] [22, 20] 115.0
State 12: [15, 14] [3, 1] 115.0
State 13: [15, 14] [6, 4] 115.0
State 14: [15, 14] [13, 11] 115.0
State 15: [15, 14] [22, 20] 115.0
State 16: [22, 21] [3, 1] 115.0
State 17: [22, 21] [6, 4] 115.0
State 18: [22, 21] [13, 11] 115.0
State 19: [22, 21] [15, 13] 115.0
State 20: [22, 21] [22, 20] 115.0
State 21: [13, 11] [11, 10] 115.0
Min Utility: 115.0

Previous State: 12
Roll: [1,6]
State 1: [3, 2] [13, 7] 109.33
State 2: [3, 2] [15, 9] 109.33
State 3: [3, 2] [22, 16] 109.33
State 4: [15, 14] [13, 7] 109.33
State 5: [15, 14] [14, 8] 109.33
State 6: [15, 14] [22, 16] 109.33
State 7: [22, 21] [13, 7] 109.33
State 8: [22, 21] [15, 9] 109.33
State 9: [22, 21] [21, 15] 109.33
State 10: [22, 21] [22, 16] 109.33
State 11: [13, 7] [7, 6] 109.33
Min Utility: 109.33333

Previous State: 12
```

```
Roll: [3,5]
State 1: [6, 3] [6, 1] 109.67
State 2: [6, 3] [13, 8] 109.67
State 3: [6, 3] [15, 10] 109.67
State 4: [13, 10] [6, 1] 109.67
State 5: [13, 10] [13, 8] 109.67
State 6: [13, 10] [15, 10] 109.67
State 7: [15, 10] [10, 7] 109.67
Min Utility: 109.66667

Previous State: 12
Roll: [2,3]
State 1: [3, 1] [6, 3] 108.67
State 2: [3, 1] [13, 10] 108.67
State 3: [6, 4] [6, 3] 108.67
State 4: [6, 4] [13, 10] 108.67
State 5: [13, 11] [6, 3] 108.67
State 6: [13, 11] [11, 8] 108.67
State 7: [13, 11] [13, 10] 108.67
State 8: [15, 13] [6, 3] 108.67
State 9: [15, 13] [13, 10] 108.67
State 10: [22, 20] [6, 3] 108.67
State 11: [22, 20] [13, 10] 108.67
Min Utility: 108.66667

Previous State: 12
Roll: [1,2]
State 1: [3, 2] [3, 1] 108.0
State 2: [3, 2] [6, 4] 108.0
State 3: [3, 2] [13, 11] 108.0
State 4: [3, 2] [15, 13] 108.0
State 5: [3, 2] [22, 20] 108.0
State 6: [15, 14] [3, 1] 108.0
State 7: [15, 14] [6, 4] 108.0
State 8: [15, 14] [13, 11] 108.0
State 9: [15, 14] [22, 20] 108.0
State 10: [22, 21] [3, 1] 108.0
State 11: [22, 21] [6, 4] 108.0
State 12: [22, 21] [13, 11] 108.0
State 13: [22, 21] [15, 13] 108.0
State 14: [22, 21] [22, 20] 108.0
State 15: [6, 4] [4, 3] 108.0
State 16: [13, 11] [11, 10] 108.0
Min Utility: 108.0

Previous State: 13
Roll: [1,6]
State 1: [3, 2] [13, 7] 114.33
State 2: [3, 2] [15, 9] 114.33
State 3: [3, 2] [22, 16] 114.33
State 4: [6, 5] [13, 7] 114.33
State 5: [6, 5] [15, 9] 114.33
State 6: [6, 5] [22, 16] 114.33
State 7: [15, 14] [13, 7] 114.33
State 8: [15, 14] [14, 8] 114.33
State 9: [15, 14] [22, 16] 114.33
State 10: [22, 21] [13, 7] 114.33
State 11: [22, 21] [15, 9] 114.33
State 12: [22, 21] [21, 15] 114.33
State 13: [22, 21] [22, 16] 114.33
State 14: [13, 7] [7, 6] 114.33
Min Utility: 114.33333

Previous State: 13
Roll: [3,5]
State 1: [6, 3] [6, 1] 114.67
State 2: [6, 3] [13, 8] 114.67
State 3: [6, 3] [15, 10] 114.67
State 4: [13, 10] [6, 1] 114.67
State 5: [13, 10] [10, 5] 114.67
State 6: [13, 10] [13, 8] 114.67
State 7: [13, 10] [15, 10] 114.67
State 8: [15, 10] [10, 7] 114.67
Min Utility: 114.66667

Previous State: 13
Roll: [2,3]
State 1: [3, 1] [6, 3] 113.67
State 2: [3, 1] [13, 10] 113.67
State 3: [6, 4] [6, 3] 113.67
State 4: [6, 4] [13, 10] 113.67
State 5: [13, 11] [6, 3] 113.67
State 6: [13, 11] [11, 8] 113.67
State 7: [13, 11] [13, 10] 113.67
State 8: [15, 13] [6, 3] 113.67
State 9: [15, 13] [13, 10] 113.67
State 10: [22, 20] [6, 3] 113.67
State 11: [22, 20] [13, 10] 113.67
Min Utility: 113.66667

Previous State: 13
Roll: [1,2]
State 1: [3, 2] [3, 1] 113.0
State 2: [3, 2] [6, 4] 113.0
State 3: [3, 2] [13, 11] 113.0
State 4: [3, 2] [15, 13] 113.0
State 5: [3, 2] [22, 20] 113.0
State 6: [6, 5] [3, 1] 113.0
State 7: [6, 5] [5, 3] 113.0
```

```
State 8: [6, 5] [6, 4] 113.0
State 9: [6, 5] [13, 11] 113.0
State 10: [6, 5] [15, 13] 113.0
State 11: [6, 5] [22, 20] 113.0
State 12: [15, 14] [3, 1] 113.0
State 13: [15, 14] [6, 4] 113.0
State 14: [15, 14] [13, 11] 113.0
State 15: [15, 14] [22, 20] 113.0
State 16: [22, 21] [3, 1] 113.0
State 17: [22, 21] [6, 4] 113.0
State 18: [22, 21] [13, 11] 113.0
State 19: [22, 21] [15, 13] 113.0
State 20: [22, 21] [22, 20] 113.0
State 21: [13, 11] [11, 10] 113.0
Min Utility: 113.0

Previous State: 14
Roll: [1,6]
State 1: [3, 2] [13, 7] 111.33
State 2: [3, 2] [15, 9] 111.33
State 3: [3, 2] [22, 16] 111.33
State 4: [15, 14] [13, 7] 111.33
State 5: [15, 14] [14, 8] 111.33
State 6: [15, 14] [22, 16] 111.33
State 7: [22, 21] [13, 7] 111.33
State 8: [22, 21] [15, 9] 111.33
State 9: [22, 21] [21, 15] 111.33
State 10: [22, 21] [22, 16] 111.33
State 11: [13, 7] [7, 6] 111.33
Min Utility: 111.33333

Previous State: 14
Roll: [3,5]
State 1: [6, 3] [6, 1] 111.67
State 2: [6, 3] [13, 8] 111.67
State 3: [6, 3] [15, 10] 111.67
State 4: [13, 10] [6, 1] 111.67
State 5: [13, 10] [13, 8] 111.67
State 6: [13, 10] [15, 10] 111.67
State 7: [15, 10] [10, 7] 111.67
Min Utility: 111.66667

Previous State: 14
Roll: [2,3]
State 1: [3, 1] [6, 3] 110.67
State 2: [3, 1] [13, 10] 110.67
State 3: [6, 4] [6, 3] 110.67
State 4: [6, 4] [13, 10] 110.67
State 5: [13, 11] [6, 3] 110.67
State 6: [13, 11] [11, 8] 110.67
State 7: [13, 11] [13, 10] 110.67
State 8: [15, 13] [6, 3] 110.67
State 9: [15, 13] [13, 10] 110.67
State 10: [22, 20] [6, 3] 110.67
State 11: [22, 20] [13, 10] 110.67
Min Utility: 110.66667

Previous State: 14
Roll: [1,2]
State 1: [3, 2] [3, 1] 110.0
State 2: [3, 2] [6, 4] 110.0
State 3: [3, 2] [13, 11] 110.0
State 4: [3, 2] [15, 13] 110.0
State 5: [3, 2] [22, 20] 110.0
State 6: [15, 14] [3, 1] 110.0
State 7: [15, 14] [6, 4] 110.0
State 8: [15, 14] [13, 11] 110.0
State 9: [15, 14] [22, 20] 110.0
State 10: [22, 21] [3, 1] 110.0
State 11: [22, 21] [6, 4] 110.0
State 12: [22, 21] [13, 11] 110.0
State 13: [22, 21] [15, 13] 110.0
State 14: [22, 21] [22, 20] 110.0
State 15: [6, 4] [4, 3] 110.0
State 16: [13, 11] [11, 10] 110.0
Min Utility: 110.0

Previous State: 15
Roll: [1,6]
State 1: [3, 2] [13, 7] 109.33
State 2: [3, 2] [15, 9] 109.33
State 3: [3, 2] [22, 16] 109.33
State 4: [15, 14] [13, 7] 109.33
State 5: [15, 14] [14, 8] 109.33
State 6: [15, 14] [22, 16] 109.33
State 7: [22, 21] [13, 7] 109.33
State 8: [22, 21] [15, 9] 109.33
State 9: [22, 21] [21, 15] 109.33
State 10: [22, 21] [22, 16] 109.33
State 11: [13, 7] [7, 6] 109.33
Min Utility: 109.33333

Previous State: 15
Roll: [3,5]
State 1: [6, 3] [6, 1] 109.67
State 2: [6, 3] [13, 8] 109.67
State 3: [6, 3] [15, 10] 109.67
State 4: [13, 10] [6, 1] 109.67
State 5: [13, 10] [13, 8] 109.67
```

```
State 6: [13, 10] [15, 10] 109.67
State 7: [15, 10] [10, 7] 109.67
Min Utility: 109.66667

Previous State: 15
Roll: [2,3]
State 1: [3, 1] [6, 3] 108.67
State 2: [3, 1] [13, 10] 108.67
State 3: [6, 4] [6, 3] 108.67
State 4: [6, 4] [13, 10] 108.67
State 5: [13, 11] [6, 3] 108.67
State 6: [13, 11] [11, 8] 108.67
State 7: [13, 11] [13, 10] 108.67
State 8: [15, 13] [6, 3] 108.67
State 9: [15, 13] [13, 10] 108.67
State 10: [22, 20] [6, 3] 108.67
State 11: [22, 20] [13, 10] 108.67
Min Utility: 108.66667

Previous State: 15
Roll: [1,2]
State 1: [3, 2] [3, 1] 108.0
State 2: [3, 2] [6, 4] 108.0
State 3: [3, 2] [13, 11] 108.0
State 4: [3, 2] [15, 13] 108.0
State 5: [3, 2] [22, 20] 108.0
State 6: [15, 14] [3, 1] 108.0
State 7: [15, 14] [6, 4] 108.0
State 8: [15, 14] [13, 11] 108.0
State 9: [15, 14] [22, 20] 108.0
State 10: [22, 21] [3, 1] 108.0
State 11: [22, 21] [6, 4] 108.0
State 12: [22, 21] [13, 11] 108.0
State 13: [22, 21] [15, 13] 108.0
State 14: [22, 21] [22, 20] 108.0
State 15: [6, 4] [4, 3] 108.0
State 16: [13, 11] [11, 10] 108.0
Min Utility: 108.0

Previous State: 16
Roll: [1,6]
State 1: [3, 2] [13, 7] 111.33
State 2: [3, 2] [15, 9] 111.33
State 3: [3, 2] [22, 16] 111.33
State 4: [15, 14] [13, 7] 111.33
State 5: [15, 14] [14, 8] 111.33
State 6: [15, 14] [22, 16] 111.33
State 7: [22, 21] [13, 7] 111.33
State 8: [22, 21] [15, 9] 111.33
State 9: [22, 21] [21, 15] 111.33
State 10: [22, 21] [22, 16] 111.33
State 11: [13, 7] [7, 6] 111.33
Min Utility: 111.33333

Previous State: 16
Roll: [3,5]
State 1: [6, 3] [6, 1] 111.67
State 2: [6, 3] [13, 8] 111.67
State 3: [6, 3] [15, 10] 111.67
State 4: [13, 10] [6, 1] 111.67
State 5: [13, 10] [13, 8] 111.67
State 6: [13, 10] [15, 10] 111.67
State 7: [15, 10] [10, 7] 111.67
Min Utility: 111.66667

Previous State: 16
Roll: [2,3]
State 1: [3, 1] [6, 3] 110.67
State 2: [3, 1] [13, 10] 110.67
State 3: [6, 4] [6, 3] 110.67
State 4: [6, 4] [13, 10] 110.67
State 5: [13, 11] [6, 3] 110.67
State 6: [13, 11] [11, 8] 110.67
State 7: [13, 11] [13, 10] 110.67
State 8: [15, 13] [6, 3] 110.67
State 9: [15, 13] [13, 10] 110.67
State 10: [22, 20] [6, 3] 110.67
State 11: [22, 20] [13, 10] 110.67
Min Utility: 110.66667

Previous State: 16
Roll: [1,2]
State 1: [3, 2] [3, 1] 110.0
State 2: [3, 2] [6, 4] 110.0
State 3: [3, 2] [13, 11] 110.0
State 4: [3, 2] [15, 13] 110.0
State 5: [3, 2] [22, 20] 110.0
State 6: [15, 14] [3, 1] 110.0
State 7: [15, 14] [6, 4] 110.0
State 8: [15, 14] [13, 11] 110.0
State 9: [15, 14] [22, 20] 110.0
State 10: [22, 21] [3, 1] 110.0
State 11: [22, 21] [6, 4] 110.0
State 12: [22, 21] [13, 11] 110.0
State 13: [22, 21] [15, 13] 110.0
State 14: [22, 21] [22, 20] 110.0
State 15: [6, 4] [4, 3] 110.0
State 16: [13, 11] [11, 10] 110.0
Min Utility: 110.0
```

```
Previous State: 17
Roll: [1,6]
State 1: [3, 2] [13, 7] 111.33
State 2: [3, 2] [15, 9] 111.33
State 3: [3, 2] [22, 16] 111.33
State 4: [15, 14] [13, 7] 111.33
State 5: [15, 14] [14, 8] 111.33
State 6: [15, 14] [22, 16] 111.33
State 7: [22, 21] [13, 7] 111.33
State 8: [22, 21] [15, 9] 111.33
State 9: [22, 21] [21, 15] 111.33
State 10: [22, 21] [22, 16] 111.33
State 11: [13, 7] [7, 6] 111.33
Min Utility: 111.33333

Previous State: 17
Roll: [3,5]
State 1: [6, 3] [6, 1] 111.67
State 2: [6, 3] [13, 8] 111.67
State 3: [6, 3] [15, 10] 111.67
State 4: [13, 10] [6, 1] 111.67
State 5: [13, 10] [13, 8] 111.67
State 6: [13, 10] [15, 10] 111.67
State 7: [15, 10] [10, 7] 111.67
Min Utility: 111.66667

Previous State: 17
Roll: [2,3]
State 1: [3, 1] [6, 3] 110.67
State 2: [3, 1] [13, 10] 110.67
State 3: [6, 4] [6, 3] 110.67
State 4: [6, 4] [13, 10] 110.67
State 5: [13, 11] [6, 3] 110.67
State 6: [13, 11] [11, 8] 110.67
State 7: [13, 11] [13, 10] 110.67
State 8: [15, 13] [6, 3] 110.67
State 9: [15, 13] [13, 10] 110.67
State 10: [22, 20] [6, 3] 110.67
State 11: [22, 20] [13, 10] 110.67
Min Utility: 110.66667

Previous State: 17
Roll: [1,2]
State 1: [3, 2] [3, 1] 110.0
State 2: [3, 2] [6, 4] 110.0
State 3: [3, 2] [13, 11] 110.0
State 4: [3, 2] [15, 13] 110.0
State 5: [3, 2] [22, 20] 110.0
State 6: [15, 14] [3, 1] 110.0
State 7: [15, 14] [6, 4] 110.0
State 8: [15, 14] [13, 11] 110.0
State 9: [15, 14] [22, 20] 110.0
State 10: [22, 21] [3, 1] 110.0
State 11: [22, 21] [6, 4] 110.0
State 12: [22, 21] [13, 11] 110.0
State 13: [22, 21] [15, 13] 110.0
State 14: [22, 21] [22, 20] 110.0
State 15: [6, 4] [4, 3] 110.0
State 16: [13, 11] [11, 10] 110.0
Min Utility: 110.0

Previous State: 18
Roll: [1,6]
State 1: [3, 2] [13, 7] 116.33
State 2: [3, 2] [15, 9] 116.33
State 3: [3, 2] [22, 16] 116.33
State 4: [6, 5] [13, 7] 116.33
State 5: [6, 5] [15, 9] 116.33
State 6: [6, 5] [22, 16] 116.33
State 7: [15, 14] [13, 7] 116.33
State 8: [15, 14] [14, 8] 116.33
State 9: [15, 14] [22, 16] 116.33
State 10: [22, 21] [13, 7] 116.33
State 11: [22, 21] [15, 9] 116.33
State 12: [22, 21] [21, 15] 116.33
State 13: [22, 21] [22, 16] 116.33
State 14: [13, 7] [7, 6] 116.33
Min Utility: 116.33333

Previous State: 18
Roll: [3,5]
State 1: [6, 3] [6, 1] 116.67
State 2: [6, 3] [13, 8] 116.67
State 3: [6, 3] [15, 10] 116.67
State 4: [13, 10] [6, 1] 116.67
State 5: [13, 10] [10, 5] 116.67
State 6: [13, 10] [13, 8] 116.67
State 7: [13, 10] [15, 10] 116.67
State 8: [15, 10] [10, 7] 116.67
Min Utility: 116.66667

Previous State: 18
Roll: [2,3]
State 1: [3, 1] [6, 3] 115.67
State 2: [3, 1] [13, 10] 115.67
State 3: [6, 4] [6, 3] 115.67
State 4: [6, 4] [13, 10] 115.67
State 5: [13, 11] [6, 3] 115.67
```

```
State 6: [13, 11] [11, 8] 115.67
State 7: [13, 11] [13, 10] 115.67
State 8: [15, 13] [6, 3] 115.67
State 9: [15, 13] [13, 10] 115.67
State 10: [22, 20] [6, 3] 115.67
State 11: [22, 20] [13, 10] 115.67
Min Utility: 115.66667

Previous State: 18
Roll: [1,2]
State 1: [3, 2] [3, 1] 115.0
State 2: [3, 2] [6, 4] 115.0
State 3: [3, 2] [13, 11] 115.0
State 4: [3, 2] [15, 13] 115.0
State 5: [3, 2] [22, 20] 115.0
State 6: [6, 5] [3, 1] 115.0
State 7: [6, 5] [5, 3] 115.0
State 8: [6, 5] [6, 4] 115.0
State 9: [6, 5] [13, 11] 115.0
State 10: [6, 5] [15, 13] 115.0
State 11: [6, 5] [22, 20] 115.0
State 12: [15, 14] [3, 1] 115.0
State 13: [15, 14] [6, 4] 115.0
State 14: [15, 14] [13, 11] 115.0
State 15: [15, 14] [22, 20] 115.0
State 16: [22, 21] [3, 1] 115.0
State 17: [22, 21] [6, 4] 115.0
State 18: [22, 21] [13, 11] 115.0
State 19: [22, 21] [15, 13] 115.0
State 20: [22, 21] [22, 20] 115.0
State 21: [13, 11] [11, 10] 115.0
Min Utility: 115.0

Previous State: 19
Roll: [1,6]
State 1: [3, 2] [13, 7] 113.33
State 2: [3, 2] [15, 9] 113.33
State 3: [3, 2] [22, 16] 113.33
State 4: [15, 14] [13, 7] 113.33
State 5: [15, 14] [14, 8] 113.33
State 6: [15, 14] [22, 16] 113.33
State 7: [22, 21] [13, 7] 113.33
State 8: [22, 21] [15, 9] 113.33
State 9: [22, 21] [21, 15] 113.33
State 10: [22, 21] [22, 16] 113.33
State 11: [13, 7] [7, 6] 113.33
Min Utility: 113.33333

Previous State: 19
Roll: [3,5]
State 1: [6, 3] [6, 1] 113.67
State 2: [6, 3] [13, 8] 113.67
State 3: [6, 3] [15, 10] 113.67
State 4: [13, 10] [6, 1] 113.67
State 5: [13, 10] [13, 8] 113.67
State 6: [13, 10] [15, 10] 113.67
State 7: [15, 10] [10, 7] 113.67
Min Utility: 113.66667

Previous State: 19
Roll: [2,3]
State 1: [3, 1] [6, 3] 112.67
State 2: [3, 1] [13, 10] 112.67
State 3: [6, 4] [6, 3] 112.67
State 4: [6, 4] [13, 10] 112.67
State 5: [13, 11] [6, 3] 112.67
State 6: [13, 11] [11, 8] 112.67
State 7: [13, 11] [13, 10] 112.67
State 8: [15, 13] [6, 3] 112.67
State 9: [15, 13] [13, 10] 112.67
State 10: [22, 20] [6, 3] 112.67
State 11: [22, 20] [13, 10] 112.67
Min Utility: 112.66667

Previous State: 19
Roll: [1,2]
State 1: [3, 2] [3, 1] 112.0
State 2: [3, 2] [6, 4] 112.0
State 3: [3, 2] [13, 11] 112.0
State 4: [3, 2] [15, 13] 112.0
State 5: [3, 2] [22, 20] 112.0
State 6: [15, 14] [3, 1] 112.0
State 7: [15, 14] [6, 4] 112.0
State 8: [15, 14] [13, 11] 112.0
State 9: [15, 14] [22, 20] 112.0
State 10: [22, 21] [3, 1] 112.0
State 11: [22, 21] [6, 4] 112.0
State 12: [22, 21] [13, 11] 112.0
State 13: [22, 21] [15, 13] 112.0
State 14: [22, 21] [22, 20] 112.0
State 15: [6, 4] [4, 3] 112.0
State 16: [13, 11] [11, 10] 112.0
Min Utility: 112.0

Previous State: 20
Roll: [1,6]
State 1: [3, 2] [13, 7] 114.33
State 2: [3, 2] [15, 9] 114.33
State 3: [3, 2] [22, 16] 114.33
```

```
State 4: [15, 14] [13, 7] 114.33
State 5: [15, 14] [14, 8] 114.33
State 6: [15, 14] [22, 16] 114.33
State 7: [22, 21] [13, 7] 114.33
State 8: [22, 21] [15, 9] 114.33
State 9: [22, 21] [21, 15] 114.33
State 10: [22, 21] [22, 16] 114.33
State 11: [13, 7] [7, 6] 114.33
Min Utility: 114.33333

Previous State: 20
Roll: [3,5]
State 1: [6, 3] [6, 1] 114.67
State 2: [6, 3] [13, 8] 114.67
State 3: [6, 3] [15, 10] 114.67
State 4: [6, 3] [22, 17] 114.67
State 5: [13, 10] [6, 1] 114.67
State 6: [13, 10] [13, 8] 114.67
State 7: [13, 10] [15, 10] 114.67
State 8: [13, 10] [22, 17] 114.67
State 9: [15, 10] [10, 7] 114.67
State 10: [22, 17] [17, 14] 114.67
Min Utility: 114.66667

Previous State: 20
Roll: [2,3]
State 1: [3, 1] [6, 3] 113.67
State 2: [3, 1] [13, 10] 113.67
State 3: [6, 4] [6, 3] 113.67
State 4: [6, 4] [13, 10] 113.67
State 5: [13, 11] [6, 3] 113.67
State 6: [13, 11] [11, 8] 113.67
State 7: [13, 11] [13, 10] 113.67
State 8: [15, 13] [6, 3] 113.67
State 9: [15, 13] [13, 10] 113.67
State 10: [22, 20] [6, 3] 113.67
State 11: [22, 20] [13, 10] 113.67
State 12: [22, 20] [20, 17] 113.67
Min Utility: 113.66667

Previous State: 20
Roll: [1,2]
State 1: [3, 2] [3, 1] 113.0
State 2: [3, 2] [6, 4] 113.0
State 3: [3, 2] [13, 11] 113.0
State 4: [3, 2] [15, 13] 113.0
State 5: [3, 2] [22, 20] 113.0
State 6: [15, 14] [3, 1] 113.0
State 7: [15, 14] [6, 4] 113.0
State 8: [15, 14] [13, 11] 113.0
State 9: [15, 14] [22, 20] 113.0
State 10: [22, 21] [3, 1] 113.0
State 11: [22, 21] [6, 4] 113.0
State 12: [22, 21] [13, 11] 113.0
State 13: [22, 21] [15, 13] 113.0
State 14: [22, 21] [22, 20] 113.0
State 15: [6, 4] [4, 3] 113.0
State 16: [13, 11] [11, 10] 113.0
Min Utility: 113.0

Previous State: 21
Roll: [1,6]
State 1: [3, 2] [13, 7] 113.33
State 2: [3, 2] [15, 9] 113.33
State 3: [3, 2] [22, 16] 113.33
State 4: [15, 14] [13, 7] 113.33
State 5: [15, 14] [14, 8] 113.33
State 6: [15, 14] [22, 16] 113.33
State 7: [22, 21] [13, 7] 113.33
State 8: [22, 21] [15, 9] 113.33
State 9: [22, 21] [21, 15] 113.33
State 10: [22, 21] [22, 16] 113.33
State 11: [13, 7] [7, 6] 113.33
Min Utility: 113.33333

Previous State: 21
Roll: [3,5]
State 1: [6, 3] [6, 1] 113.67
State 2: [6, 3] [13, 8] 113.67
State 3: [6, 3] [15, 10] 113.67
State 4: [13, 10] [6, 1] 113.67
State 5: [13, 10] [13, 8] 113.67
State 6: [13, 10] [15, 10] 113.67
State 7: [15, 10] [10, 7] 113.67
Min Utility: 113.66667

Previous State: 21
Roll: [2,3]
State 1: [3, 1] [6, 3] 112.67
State 2: [3, 1] [13, 10] 112.67
State 3: [6, 4] [6, 3] 112.67
State 4: [6, 4] [13, 10] 112.67
State 5: [13, 11] [6, 3] 112.67
State 6: [13, 11] [11, 8] 112.67
State 7: [13, 11] [13, 10] 112.67
State 8: [15, 13] [6, 3] 112.67
State 9: [15, 13] [13, 10] 112.67
State 10: [22, 20] [6, 3] 112.67
State 11: [22, 20] [13, 10] 112.67
```

```
Min Utility: 112.66667

Previous State: 21
Roll: [1,2]
State 1: [3, 2] [3, 1] 112.0
State 2: [3, 2] [6, 4] 112.0
State 3: [3, 2] [13, 11] 112.0
State 4: [3, 2] [15, 13] 112.0
State 5: [3, 2] [22, 20] 112.0
State 6: [15, 14] [3, 1] 112.0
State 7: [15, 14] [6, 4] 112.0
State 8: [15, 14] [13, 11] 112.0
State 9: [15, 14] [22, 20] 112.0
State 10: [22, 21] [3, 1] 112.0
State 11: [22, 21] [6, 4] 112.0
State 12: [22, 21] [13, 11] 112.0
State 13: [22, 21] [15, 13] 112.0
State 14: [22, 21] [22, 20] 112.0
State 15: [6, 4] [4, 3] 112.0
State 16: [13, 11] [11, 10] 112.0
Min Utility: 112.0
```