Modeling Under Uncertainty HW1

1a $P(H \cap B^c \cap S^c)$

b $P(B \cap H \cap S^c)$

c $P(B \cup H \cup S)$

d $P((B \cap H) \cup (B \cap S) \cup (H \cap S))$

e $P(B \cap H \cap S)$

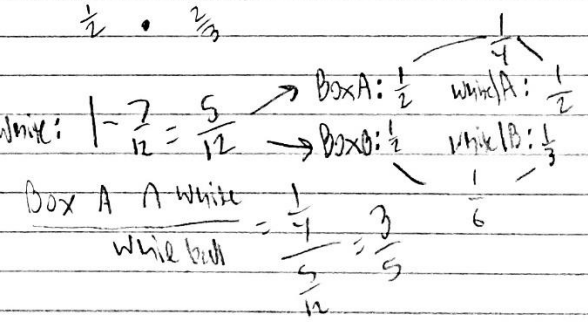
f $P(B^c \cap H^c \cap S^c)$

g $1 - P((B \cap H) \cup (B \cap S) \cup (H \cap S))$

h $1 - P(B \cap H \cap S)$

3a $\frac{7}{12}$ = Box A and Red | Box A $\frac{1}{2} \cdot \frac{1}{2}$ + Box B and Red | Box B $\frac{1}{2} \cdot \frac{2}{3}$

b $\frac{3}{5}$ Probability ball is white: $1 - \frac{7}{12} = \frac{5}{12}$



$\frac{\text{Box A and White}}{\text{White both}} = \frac{\frac{1}{4}}{\frac{5}{12}} = \frac{3}{5}$

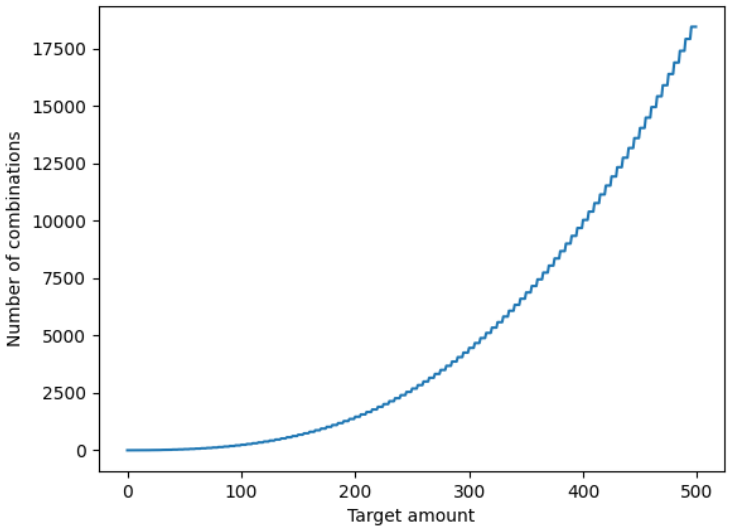
```python
import matplotlib.pyplot as plt

def count_combinations(coins_list, amount, max_coin):
    total = 0
    if amount == 0:
        total += 1
    for coin in coins_list:
        if coin <= amount and coin <= max_coin:
            total += count_combinations(coins_list, amount - coin, coin)
    return total

if __name__ == "__main__":
    print("2) b. " + str(count_combinations([1,5,10,25], 213, 1000)))
    print("   c. " + str(count_combinations([1,5,10], 213, 1000)))
    x = []
    y = []
    for val in range(500):
        x.append(val)
        y.append(count_combinations([1,5,10,25], val, 1000))
    plt.xlabel("Target amount")
    plt.ylabel("Number of combinations")
    plt.plot(x, y)
    plt.savefig("fig2d")
```

```
(.venv) C:\Users\glase\CornellTech\Modeling-Under-Uncertainty\hw1>python hw1.py
2) b. 1670
   c. 484
```

2c. 484 / 1670 = ~0.29



2d.