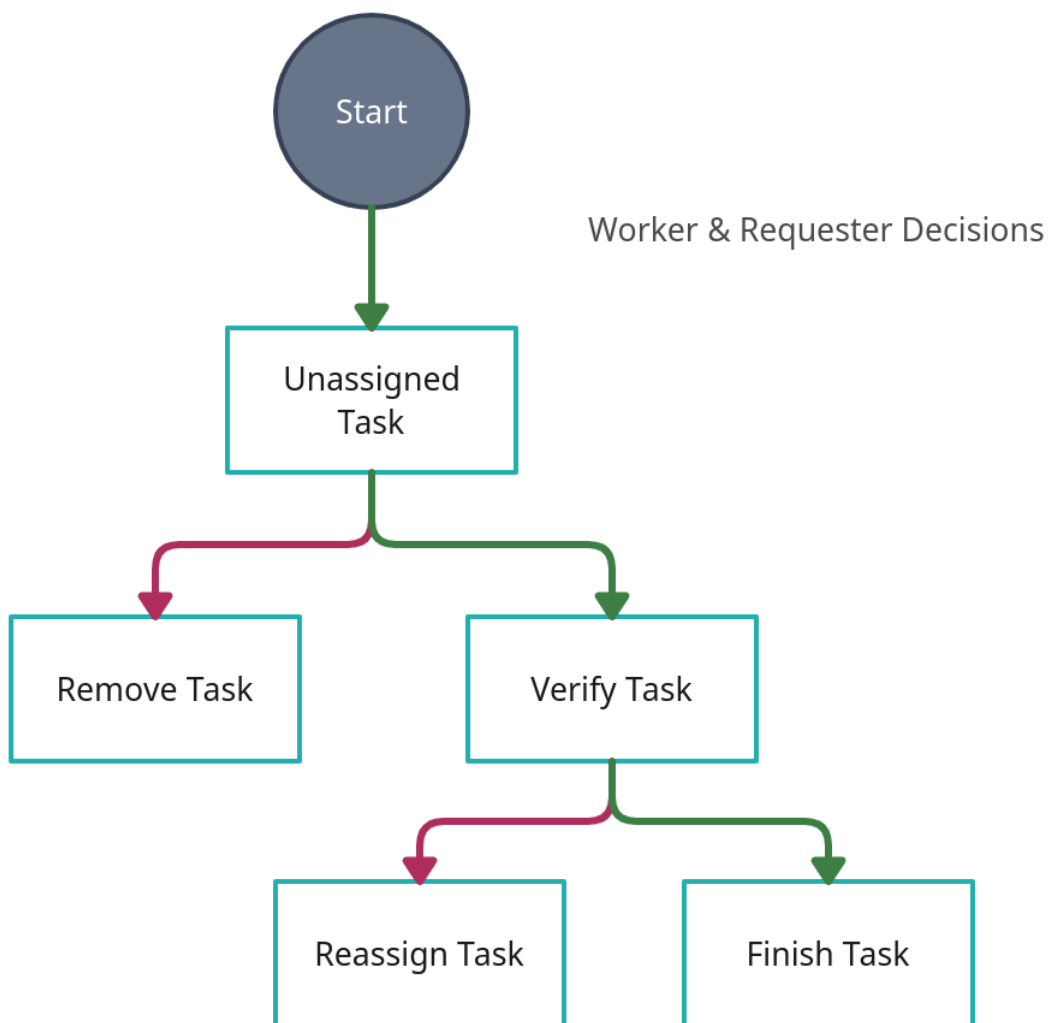


Completed tasks

The number of tasks performed according to the level of well-behaved participants



```
import random as rand
```

```
def calculateRequesterBalance(decisionProbability, taskCount):
    if decisionProbability < 0 or decisionProbability > 1:
        raise ValueError(decisionProbability, "Args must be between 0 and 1")
    else:
        result = 0
        for i in range(0, taskCount):
            goodBehaviorProbability = rand.random()
            waitUntilTaskDoneProbability = goodBehaviorProbability / 10
            finishTaskProbability = goodBehaviorProbability * 9 / 10

            # Requester waits until done task
            if waitUntilTaskDoneProbability <= decisionProbability:

                # Finish task by requester
                if finishTaskProbability <= decisionProbability:
                    result = result + 1
                # Unfinish task

        return result / taskCount

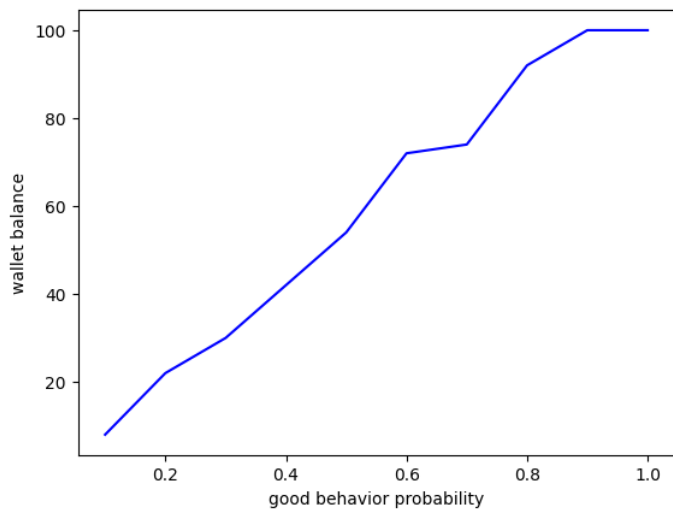
return
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
def showFinishedTasksCount(testCount):
    requesterInitialBalance = rand.randint(10, 20)
    probabilities = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
```

```
outputs = [calculateRequesterBalance(i, testCount) * testCount for i in probabilities]
plt.plot(probabilities, outputs, 'b-')
plt.xlabel("good behavior probability")
plt.ylabel("wallet balance")
plt.show()
```

```
showFinishedTasksCount(100)
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 3:02 PM

