

**12 nuggets all of equal weight except 1. The odd one out could be lighter or heavier. Algorithm to determine which one.**

### Function to store nugget weights in dictionary

*Assume 10 of the nuggets weigh 1 unit*

```
In [129]: def nuggets_init (odd_nugget_no, odd_nugget_weight):  
    nugget = {}  
    f = lambda nugget, odd_nugget_no, odd_nugget_weight: odd_nugget_weight if  
    (nugget == odd_nugget_no) else 1  
    nuggets = {nugget: f(nugget, odd_nugget_no, odd_nugget_weight) for nugget in  
    range(1,13)}  
    return(nuggets)
```

### Function to determine which nugget is the odd one out

```

In [138]: def odd_one_out (n):
            # Weight nuggets 1...4 vs. 5...8
            # Scenario 1: 1...4 equal to 5...8
            # Scenario 2: 1...4 heavier than 5...8
            # Scenario 3: 1...4 lighter than 5...8

            if (n[1]+n[2]+n[3]+n[4]) == (n[5]+n[6]+n[7]+n[8]): #Scenario 1, weighing
1. Nuggets 9-12 contain the odd one out.
                if (n[9] == n[10]): #Scenario 1, weighing 2. Nuggets 11 or 12 are th
e odd one out
                    if (n[11] == n[1]): #Scenario 1, weighing 3. Nugget 12 is the od
d one out.
                        return (12, n[12])
                    else: #Scenario 1, weighing 3. Nugget 11 is the odd one out.
                        return (11, n[11])
                else: #Scenario 1, weighing 2. Nuggets 9 or 10 are the odd one out.
                    if (n[9] == n[1]): #Scenario 1, weighing 3. Nugget 10 is the odd
one out.
                        return (10, n[10])
                    else: #Scenario 1, weighing 3. Nugget 9 is the odd one out.
                        return (9, n[9])
            elif (n[1]+n[2]+n[3]+n[4]) > (n[5]+n[6]+n[7]+n[8]): #Scenario 2, weighin
g 1. Nuggets 1-8 contain the odd one out.
                if (n[1]+n[5]+n[6]+n[7]) > (n[8]+n[9]+n[10]+n[11]): #Scenario 2, wei
ghing 2. Nuggets 1 or 8 are the odd one out
                    if (n[1] == n[9]): #Scenario 2, weighing 3. Nugget 8 is the odd
one out.
                        return (8, n[10])
                    else: #Scenario 2, weighing 3. Nugget 1 is the odd one out.
                        return (1, n[1])
                elif (n[1]+n[5]+n[6]+n[7]) < (n[8]+n[9]+n[10]+n[11]): #Scenario 2, w
eighing 2. Nuggets 5...7 contain the odd one out.
                    if (n[5] < n[6]): #Scenario 2, weighing 3. Nugget 5 is the odd
one out.
                        return (5, n[5])
                    elif (n[6] < n[5]): #Scenario 2, weighing 3. Nugget 6 is the odd
one out.
                        return (6, n[6])
                    elif (n[6] == n[5]): #Scenario 2, weighing 3. Nugget 7 is the odd
one out.
                        return (7, n[7])
                elif (n[1]+n[5]+n[6]+n[7]) == (n[8]+n[9]+n[10]+n[11]): #Scenario 2,
weighing 2. Nuggets 2...4 contain the odd one out.
                    if (n[2] > n[3]): #Scenario 2, weighing 3. Nugget 2 is the odd
one out.
                        return (2, n[2])
                    elif (n[3] > n[2]): #Scenario 2, weighing 3. Nugget 3 is the odd
one out.
                        return (3, n[3])
                    elif (n[2] == n[3]): #Scenario 2, weighing 3. Nugget 4 is the odd
one out.
                        return (4, n[4])
                elif (n[1]+n[2]+n[3]+n[4]) < (n[5]+n[6]+n[7]+n[8]): #Scenario 3, weighin
g 1. Nuggets 1-8 contain the odd one out.
                    if (n[1]+n[5]+n[6]+n[7]) < (n[8]+n[9]+n[10]+n[11]): #Scenario 2,
weighing 2. Nuggets 1 or 8 are the odd one out

```

```

        if (n[1] == n[9]):    #Scenario 2, weighing 3. Nugget 8 is the
odd one out.
            return (8, n[10])
        else:    #Scenario 2, weighing 3. Nugget 1 is the odd one out.
            return (1, n[1])
        elif (n[1]+n[5]+n[6]+n[7]) > (n[8]+n[9]+n[10]+n[11]):    #Scenario
2, weighing 2. Nuggets 5...7 contain the odd one out.
            if (n[5] > n[6]):    #Scenario 2, weighing 3. Nugget 5 is the
odd one out.
                return (5, n[5])
            elif (n[6] > n[5]):    #Scenario 2, weighing 3. Nugget 6 is the
odd one out.
                return (6, n[6])
            elif (n[6] == n[5]):    #Scenario 2, weighing 3. Nugget 7 is the
odd one out.
                return (7, n[7])
        elif (n[1]+n[5]+n[6]+n[7]) == (n[8]+n[9]+n[10]+n[11]):    #Scenario
2, weighing 2. Nuggets 2...4 contain the odd one out.
            if (n[2] < n[3]):    #Scenario 2, weighing 3. Nugget 2 is the
odd one out.
                return (2, n[2])
            elif (n[3] < n[2]):    #Scenario 2, weighing 3. Nugget 3 is the
odd one out.
                return (3, n[3])
            elif (n[2] == n[3]):    #Scenario 2, weighing 3. Nugget 4 is the
odd one out.
                return (4, n[4])

```

## Run a test ...

In [161]: `import random`

```

for n in range (1,13):
    w = float(f'{random.uniform(1,100):0.2f}')
    nuggets = nuggets_init(n,w)    # Assign nugget number "n" a weight of "w"
    (odd_nugget, odd_nugget_weight) = odd_one_out(nuggets)
    print (f'Nugget #{odd_nugget} is the odd one out with a weight of {odd_nug
get_weight}')

```

```

Nugget #1 is the odd one out with a weight of 52.74
Nugget #2 is the odd one out with a weight of 39.47
Nugget #3 is the odd one out with a weight of 33.79
Nugget #4 is the odd one out with a weight of 40.25
Nugget #5 is the odd one out with a weight of 36.66
Nugget #6 is the odd one out with a weight of 39.42
Nugget #7 is the odd one out with a weight of 30.11
Nugget #8 is the odd one out with a weight of 1
Nugget #9 is the odd one out with a weight of 89.34
Nugget #10 is the odd one out with a weight of 73.1
Nugget #11 is the odd one out with a weight of 6.53
Nugget #12 is the odd one out with a weight of 18.9

```

In [ ]: