ALGORITHM CREATION AND ANALYSIS II (MATRYOSHKA)

MATRYOSHKA NESTING DOLLS

Use what you learned by completing Algorithm Creation and Analysis I to help you complete this assignment.

Premise:

On a table there are **24 sets of** Matryoshka doll sets. A Matryoshka set consists of wooden figures that separate in the middle to reveal a smaller figure of the same sort nested inside, which also has a nested figure, continuing the nesting down until the smallest figure is reached. There are several nested figures in a set.

Each set has a different number of nested dolls within it, with the highest nested amount of 50, and the smallest nested amount of 3.

PART ONE: COUNT ALL

Create an algorithm that will **count how many dolls there are in total from all sets**. Write your algorithm in pseudocode:

```
CountAll()
    amount = 0

    foreach set in dollSets
        amount = amount + set.amount
```

Analyze your algorithm using big O notation.

- Worst runtime: `O(1) when there is one set of dolls`
- Best runtime: `O(n) when there are many sets of dolls`

PART TWO: LINE UP SETS

Create an algorithm that will **line the doll sets up from the one with the smallest nested amount to the largest**.

Write your algorithm in pseudocode:

```
LineUpSets()
    n = dollSets.amount
    for i = 0 to i = n - 1
        for j = 0 to j = n - i - 1
            if dollSets[j].amount > dollSets[j+1].amount
                swap dollSets[j] with dollSets[j+i]
```

Analyze your algorithm using big O notation.

- Worst runtime: O(n) when the dolls are already sorted
- Best runtime: O(n^2)

PART THREE: FIND UNIQUE IN SET

One of the dolls is unique and has the artist's signature on it. **Create an algorithm that searches through all dolls in all sets until it finds the one with the signature** (and then stops).

Write your algorithm in pseudocode:

```
FindSignature(signature)
    foreach set in dollSets
        foreach doll in set
            if doll.signature == signature
                return doll
```

Analyze your algorithm using big O notation.

- Worst runtime: O(1) when the signature matches the first doll in the first set
- Best runtime: O(n^2) when the signature is in the last doll of the last set

PART FOUR: SUMMARY

Compare your analysis of questions 1 and 3.

What conclusions can you draw based on what you learned?

The CountAll and FindSignature algorithms have the same best-case time, but differ in worst-case time because the FindSignature alogorithm has a nested loop.

HELPFUL RESOURCES:

- Asymptotic Notation
- The Idea Behind Big O Notation