

Problem 3B

1. This solution achieves $O(\log n)$ time because for each iteration of the while loop, it multiplies i by 2, so in total it would take $O(\log N)$ time to find the index.

```
def ProblemThree(data, value):
```

```
    i = 1
```

```
    compare = 0
```

```
    count = 1
```

```
    while count == 1:
```

```
        if value == data.getValue(i):
```

```
            compare += 1
```

```
            print("compares:" + str(compare))
```

```
            return value
```

```
        elif value > data.getValue(i):
```

```
            compare += 1
```

```
            i *= 2 // This multiplies the iterations by two every time
```

```
        elif value < data.getValue(i):
```

```
            compare += 1
```

```
            i -= 1
```

```
    else:
```

```
        print("value not Found!")
```

At the beginning of the while loop, $i = 1$, and after every iteration, it is multiplied by two, until the `getValue(i)` is greater than the value, then it decrements the i by one, until it finds the matching element. Because the decrements take constant time, they can be ignored, giving a final time complexity of $O(\log N)$.