

Problem 1

```
In [ ]: # (a) divide vector into two halves until size = 1
def divide(v, left, right):
    if left == right: # base case
        return

    mid = (left + right) // 2
    divide(v, left, mid)
    divide(v, mid + 1, right)
```

```
In [ ]: # (b) divide vector into m
def divide_m(v, left, right, m):
    if right - left + 1 <= 1:
        return

    length = right - left + 1
    part = length // m

    for i in range(m):
        new_left = left + i * part

        # Last part takes remaining elements
        if i == m - 1:
            new_right = right
        else:
            new_right = new_left + part - 1

    divide_m(v, new_left, new_right, m)
```

```
In [ ]: # (c) find the position of unique 1
def find_one(v, left, right):
    if left == right:
        if v[left] == 1:
            return left
        return -1

    mid = (left + right) // 2

    pos = find_one(v, left, mid)
    if pos != -1:
        return pos

    return find_one(v, mid + 1, right)
```

```
In [ ]: # test
v = [0,0,0,0,1,0,0]
divide(v, 0, len(v)-1)
print("Position of 1:", find_one(v, 0, len(v)-1))
```

Position of 1: 4

Binary divide: $T(n) = 2T(n/2) O(n)$

Ternary divide: $T(n) = 3T(n/3) O(n)$

Problem 2

```
In [7]: # basic school multiplication using arrays
def multiply_basic(x, y):
    X = list(map(int, str(x))[:-1])
    Y = list(map(int, str(y))[:-1])

    result = [0]*(len(X)+len(Y))

    for i in range(len(X)):
        for j in range(len(Y)):
            result[i+j] += X[i]*Y[j]

            # carry
            result[i+j+1] += result[i+j]//10
            result[i+j] %= 10

    while len(result)>1 and result[-1]==0:
        result.pop()

    result.reverse()
    return int("".join(map(str,result)))
```

```
In [8]: # Large number multiplication (string input)
def multiply_large(a, b):
    X = list(map(int, a))[:-1]
    Y = list(map(int, b))[:-1]

    result = [0]*(len(X)+len(Y))

    for i in range(len(X)):
        for j in range(len(Y)):
            result[i+j] += X[i]*Y[j]
            result[i+j+1] += result[i+j]//10
            result[i+j] %= 10

    while len(result)>1 and result[-1]==0:
        result.pop()

    result.reverse()
    return "".join(map(str,result))

print("123 * 456 =", multiply_basic(123,456))
print("Big multiply =", multiply_large("999999999", "888888888"))
```

123 * 456 = 56088

Big multiply = 888888887111111112

Normal multiplication: $T(n) = 4T(n/2) \ O(n^2)$

Karatsuba: $T(n) = 3T(n/2) \ O(n^{1.585})$

Sum 1..n: $n(n+1)/2 \ v = n \ w = n+1$