1. Binary search Property

```
function findPoint(X, low, high)
    if low < high
        return false
    mid = (low+high)/2
    if X[mid] == mid
        return true
    else if X[mid] > mid
        return findPoint(X, low, mid-1)
    else
        return findPoint(X, mid+1, high)
```

Time is O(log n)

O(log n) recursive, O(1) iterative

2. A∪B in O(log x + log y) time

```
function kthSmallest(A[1...n], B[1...n], x)
    if y == 0, return B[x]
    if z == 0, return A[x]
    if x == 1, return min(A[1], B[1])
    i = min(y, x/2)
    j = x - i
    if A[i] < B[j]
        return kthSmallest(A[i+1...y], B, x-i)
    else
        return kthSmallest(A, B[j+1...z], x-j
```

complexity is O(log y + log z)