Problem 1 Analysis

Show O() by mathematical analysis:

Linear Search

```
for (int i=0; i<n; i++) {
    if (val==a[i])
        return i;
}
return -1;
```

$\Rightarrow O_b + \sum_{i=0}^{n-1} (O_i + POs)$ Let $O_i + POs = O_{is}$

$= \sum_{i=x}^{y} 1 = (y-x)+1$

$\Rightarrow O_b + \sum_{i=0}^{n-1} O_{is}$

$\Rightarrow O_b + O_{is}(n-x-0+1)$

$\Rightarrow \underbrace{O_{is}}_{c'}(n) + \underbrace{O_b}_{c} \Rightarrow \boxed{c'N+C}$ behaves like an order N function

$\therefore$ $\boxed{\text{Linear Search B } O(N)}$

Binary Search

```
int low=0;
int high=0;
    do {
        int middle=(high+low)/2;
        if(val==a[middle]) return middle;
```

```
else if(val > a[middle]) low=middle+1;
else high=middle-1;
} while(low <= high)
return -1;
```

$$\Rightarrow O_b + \sum_{i=0}^{\log_2(n)} (O_i + PO_s)$$

$$\Rightarrow O_b + \sum_{i=0}^{\log_2(n)} O_{is}$$

$$\Rightarrow O_b + O_{is} \sum_{i=0}^{\log_2(n)} 1$$

$$\Rightarrow O_b + O_{is}(1 + \log_2(n))$$

$$\Rightarrow \underbrace{O_b + O_{is}}_{c} + \underbrace{O_{is}}_{c'} \log_2(n)$$

$$\Rightarrow c + c' \log_2(n) = f(n)$$

$$\Rightarrow \boxed{f(n) = O(\log(N))}$$

Array size $= n$

After one;

size $= \frac{n}{2}$

After two;

size $= \frac{n}{4}$

After k iterations

$1 = \frac{N}{2^k}$

$2^k = N$

$k = \log_2(N)$