```
int getFirst(int[] data){
    if(data.length < 1)
        throw new NoSuchElementEx();
    return data[0];
}
```

length (n)	array accesses
10	1
100	1
1000 000	1

$O(1)$ constant

```
p=0
x=2
for(int i=2; i<=n; i++){
    p=(p+1)*x;
}
```

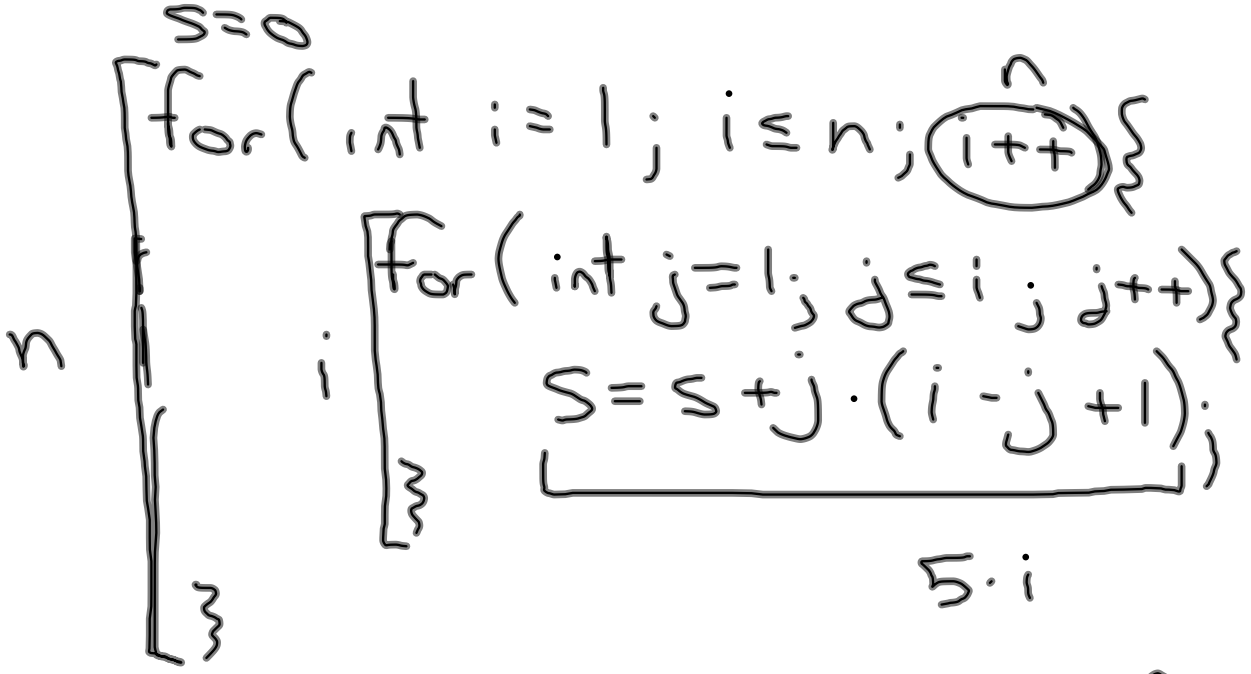
iterations of the loop $n-2+1=n-1$

additions: control loop: $n-1$

in the loop: $n-1$

multiplications: $n-1$

$3(n-1)$ $\quad O(n)$ $\quad \Omega(n)$ $\quad \Theta(n)$



$$n + \sum_{i=1}^{n} 5i = n + 5 \cdot \sum_{i=1}^{n} i$$

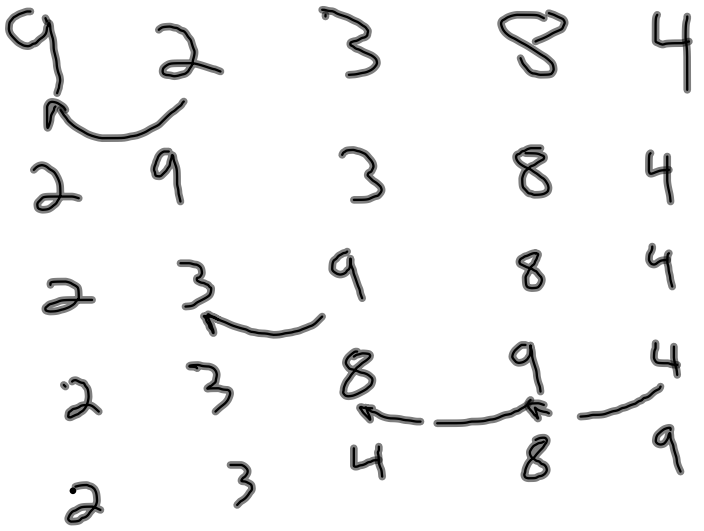
$$= n + 5\left(\frac{n(n+1)}{2}\right)$$

$$= n + 5\left(\frac{1}{2}n^2 + \frac{1}{2}n\right)$$

$$= n + \frac{5}{2}n^2 + \frac{5}{2}n = \frac{3}{2}n + \frac{5}{2}n^2$$

$$O(n^2)$$

Insertion Sort



n [for each item i at index idx
idx-1 [move i to the left as long as it is smaller than the one before

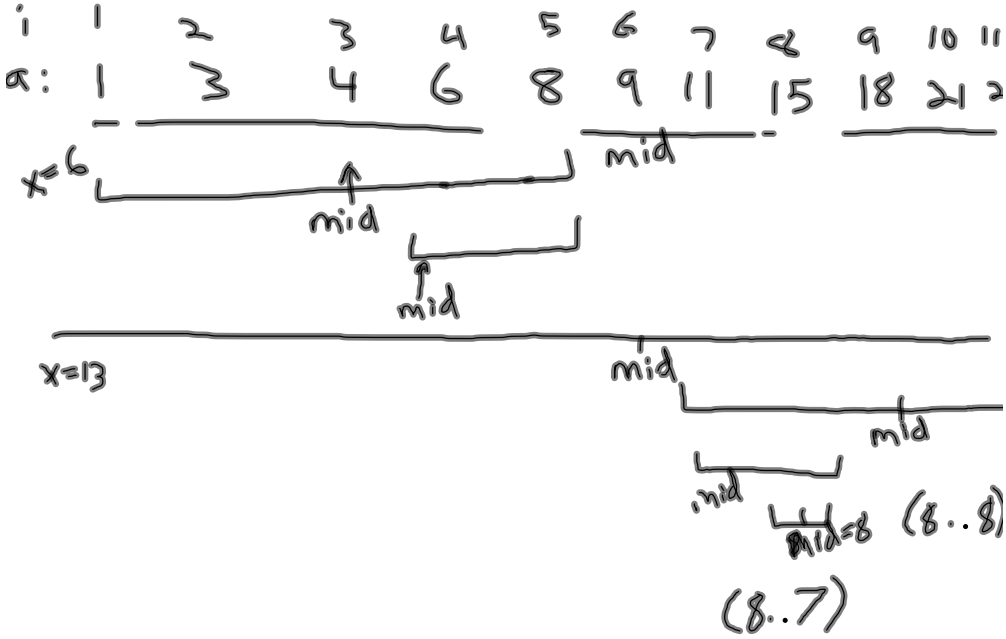
$\Theta(n^2)$

$$y = 2^x$$

$$x = \log_2 y$$

Binary Search a[1].. a[n]
search for x

```
index = 0
low = 1
high = n
while (high >= low && index = 0){
    mid = ⌊ high+low/2 ⌋
    if (a[mid] == x)
        index = mid
    if(a[mid] > x)
        high = mid - 1
    else
        low = mid + 1
}
```



start w/ k elements $1..k$

$mid = \lfloor \frac{k+1}{2} \rfloor$

$1..\lfloor \frac{k+1}{2} \rfloor - 1$ or $\lfloor \frac{k+1}{2} \rfloor + 1 \ldots k$

---

k is even mid $\frac{k}{2}$ $\underbrace{1\ 2\ 3\ 4}_{\text{mid}}$

$\underbrace{1 \ldots \frac{k}{2}-1}_{(\frac{k}{2}-1)-1+1 = \frac{k}{2}-1}$, $\frac{k}{2}$, $\underbrace{\frac{k}{2}+1 \ldots k}_{k-(\frac{k}{2}+1)+1 = k-\frac{k}{2}-1+1}$

$\frac{k}{2}$

---

k is odd mid $\frac{k+1}{2}$

$\underbrace{1 \ldots \frac{k+1}{2}-1}_{\lfloor \frac{k}{2} \rfloor}$, $\frac{k+1}{2}$, $\underbrace{\frac{k+1}{2}+1 \ldots k}_{\lfloor \frac{k}{2} \rfloor}$

$$\omega_k = 1 + \omega_{\lfloor \frac{k}{2} \rfloor}$$

$$\omega_1 = 1$$