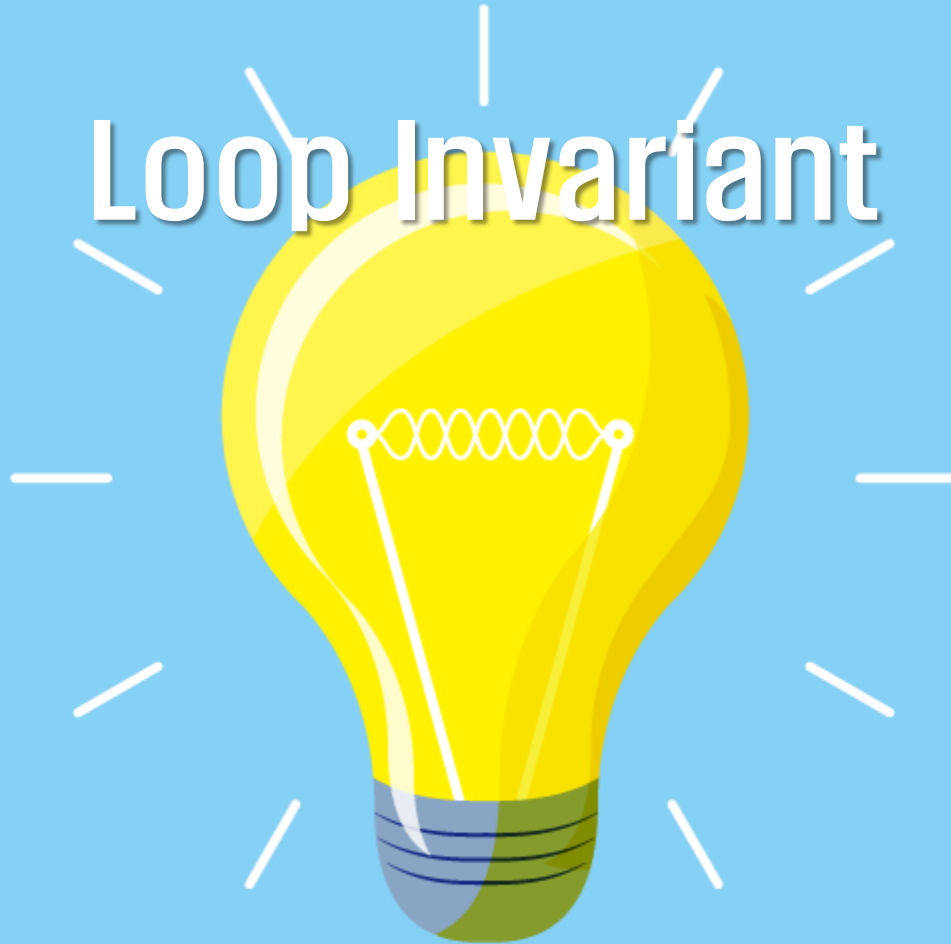


Loop Invariant



201702002 김지혜



Loop invariant?



활용 방법



활용 예시



Loop invariant ?.



Loop invariant ?.

- Invariant : 변함없는, 변치 않는
- Loop invariant
 - ‘루프 불변자’
 - 루프를 돌 동안 변하지 않는 구문 이라는 뜻
 - 루프를 사용 할 때 루프 점검에 쓰이는 기법!



활용 방법 ??



활용 방법 ??

- Loop invariant 를 이용한 증명은 3단계로 나뉜다
 - 1. Initialization
 - 루프에 진입하는 시점에 참인지?
(loop invariant에 맞는지)
 - 2. Maintenance
 - 루프 반복 직전에 참이라면,
다음 루프에서도 참인지?
($n-1$ 번째가 참이면 n 도 참)
 - 3. Termination
 - 루프가 종료되었을 때 참인지?



활용 방법 ??

- 수학적 귀납법 생각하면 좀 더 이해하기 쉽다!
 - 1. Initialization – 초기값, $n=1$ 일 때 성립하는지?
 - 2. Maintenance – $n=k$ 일 때, $k+1$ 이 성립하는지?
 - 하지만, Termination이 없다는 점에서
Loop invariant와 다르다!
→ loop invariant는 loop가 수행중인 동안에 성립



활용 예시



활용 예시

■ 선택 정렬을 보자!

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a[] = {3, 2, 54, 43, 89, 64};  
  
        int minLoc;  
        for(int i=0; i<6; i++)  
        {  
            minLoc=i;  
            for(int j=i; j<6; j++)  
            {  
                if(a[minLoc]>a[j])  
                    minLoc=j;  
            }  
            int temp=a[minLoc];  
            a[minLoc]=a[i];  
            a[i]=temp;  
        }  
        for(int i=0; i<6; i++)  
            System.out.println(a[i]);  
    }  
}
```



활용 예시

- Initialization

3	2	54	43	89	64
---	---	----	----	----	----

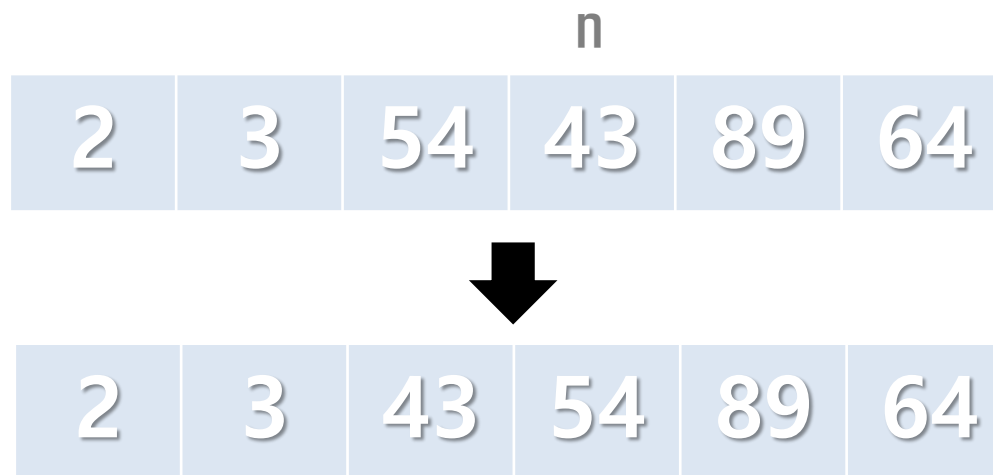
2	3	54	43	89	64
---	---	----	----	----	----

i=0일 때 최소 값[2]이 맨 앞으로 온다
→ 0번째 index 정렬!



활용 예시

▪ Maintenance



$i=n(n<6)$ 일 때 0부터 n 번째 index까지 정렬됨!



활용 예시

- Termination

2	3	43	54	64	89
---	---	----	----	----	----

끝까지 정렬이 됨 => loop invariant 성립!



감사합니다!