

단순 선택 정렬

🕒 작성일시	@2023년 6월 15일 오전 2:11
📄 강의 번호	Algorithm
📄 유형	
📎 자료	
☑ 복습	<input type="checkbox"/>
☰ Spring Framework	



선택 정렬은 말 그대로 위치에 들어갈 데이터를 찾아 선택하는 알고리즘 이다.
데이터를 '비교'하며 찾는 '비교정렬'이다.

정렬의 대상이 되는 데이터 외에 추가적인 공간을 필요로 하지 않기 때문에 '제자리 정렬'이다.

그리고 '불안정 정렬이다.'

과정

1. 주어진 리스트에서 최솟값을 찾는다.
2. 최솟값을 맨 앞자리 값과 교환한다.
3. 맨 앞 자리를 제외한 나머지 값들 중 최솟값을 찾아 위와 같은 방법을 반복한다.

장점

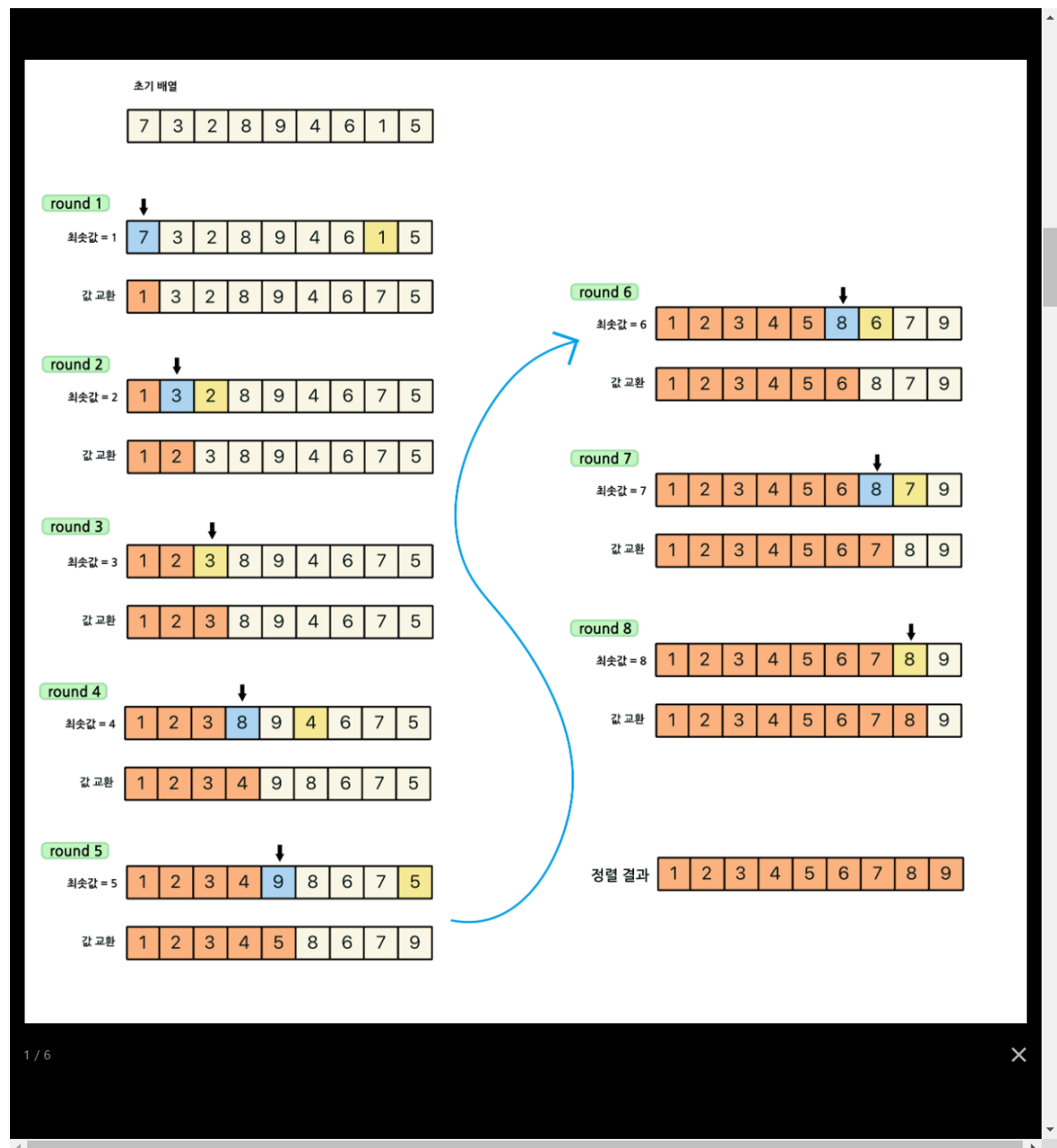
추가적인 메모리 소비가 작다.

단점

시간복잡도가 $O(N^2)$ 이다.

안정정렬이 아니다.

<https://st-lab.tistory.com/168>



불안정 정렬이란?

예시

arr = [B1, B2, C, A] (A<B<C)

B1과 B2가 서로 크거나 작은 것이 아니라는 점에 유의하자.

순서대로 순회하면서 교환하면 이러하다.

round1 : [A, B2, C, B1]

round2 : [A, B2, C, B1]

```
round3 : [A, B2, B1, C]
```

초기의 B1과 B2의 순서가 바뀐것을 볼 수 있다.
이러한 상태를 불안정 정렬이라고 한다.

예를 들어 학생을 관리할때 성적순으로 나열하되, 성적이 같으면 이름을 기준으로 정렬하고 싶다.
즉, 정렬 규칙이 다수이거나 특정 순서를 유지해야 할 문제가 될 수 있다.

예시2)

이름순 정렬

```
arr = [(가영, 60), (가희, 60), (동우, 45), (찬호, 70)]
```

그 이후 성적순 정렬

```
round1 : [(동우, 45), (가희, 60), (가영, 60), (찬호, 70)]
```

```
round2 : [(동우, 45), (가희, 60), (가영, 60), (찬호, 70)]
```

```
round3 : [(동우, 45), (가희, 60), (가영, 60), (찬호, 70)]
```

이렇게 가희보다 가영이 앞에 있어야 함에도 순서가 바뀌어 버린 것을 볼 수 있다.

구현 코드

```
static void selectionSort(int[] a, int n) {  
    for(int i=0; i<n; i++) {  
        int min = a[i];  
        for(int j=i+1; j<n; j++) {  
            if(a[j] < min) {  
                min = j;  
                swap(a,i, min)  
            }  
        }  
    }  
}
```