

코드로 저기 에 있는 것을 하나하나 구현해보자!!

//일단 이름은 들어온 배열형식의 데이터를 선택정렬을 한다는 의미로 만들었다. (명명법은 아직 더 많이 공부해야 할 것 같다.) 이 메서드의 기능은 정렬되지 않은 데이터의 집합들을 정렬해서 보내주는 것이기에 {}안에서 정렬 후 다시 배열형식으로 외부로 보낼 것이다.

```
+(NSArray*)doThisSelectionSort:(NSArray*)data{
```

```
//들어온 배열들은 대부분 고정된 배열을 쓴다. 우리는 고정된 길이가 아닌 데이터를 다루는데 좀더 편리하고,여러가지 유용한 메서드를 사용 할 수 있도록 NSMutableArray의 배열형식으로 data를 unSortedArray에 넣어서 사용 할 것이다.
```

```
NSMutableArray *unSortedArray =[[NSMutableArray alloc] initWithArray:data];
```

```
//이제 위에 있는 것을 구현해 볼 수 있도록하자!
```

```
//기존의 맨앞의 수
```

```
//기존의 맨앞의 수는 데이터의 가장 첫번째이다. 배열의 처음은 인덱스0에 있으므로(배열은 처음이 0) 정수타입으로 배열의 처음인 0의 수를 가져온다.
```

```
NSInteger i=0;
```

```
NSInteger firstValue = [unSortedArray[i] integerValue];
```

```
//비교
```

```
//이제 맨앞의 수인 0과 나머지 데이터들을 비교 해야한다.우리는 옆에있는 것(i+1)을 검사 할 것이다.
```

```
NSInteger j=i+1;
```

```
NSInteger remainderValue = [unSortedArray[j] integerValue];
```

```
//맨앞의 수와 옆에있는 수인 두번째수를 나머지를 가져와 `비교`한다.
```

```
if(firstValue>remainderValue){
```

```
//맨앞의 수의 수와 자리를 바꾼다.
```

```
[unSortedArray exchangeObjectAtIndex:i withObjectAtIndex:j]; //자리를 바꾸는 메서드를 사용!
```

```
}
```

```
return unSortedArray;
```

```
}
```

이렇게 맨앞의 수와 나머지수를 한번 바꿔보았다. 만약 정렬되지않은 데이터가 두개라면 저 메서드를 사용해도 된다.

하지만 데이터가 두개만 들어 올 일은 거의 없기 때문에 우리는 여러개의 데이터가 들어올 메서드를 만들어야 한다.

그렇기 위해서는 남아 있는 키워드를 명령어로 컴퓨터에게 알려줘야 한다.

하나하나 씩

그다음 수가 맨앞

이 두가지 키워드를 적용하자!

먼저 하나하나씩은 위에서 하나만한 비교하는 코드를 하나하나씩 마지막까지 비교하도록 만들면 된다.

```
+(NSArray*)doThisSelectionSort:(NSArray*)data{
```

```
    NSMutableArray *unSortedArray =[[NSMutableArray alloc] initWithArray:data];
```

```
    //기존의 맨앞의 수
```

```
        NSInteger i=0;
```

```
        NSInteger firstValue = [unSortedArray[i] integerValue];
```

```
    //하나하나씩
```

```
    // 옆에있는 데이터가 한개아니라 여러개라면 두번째것을 비교하고 그다음 세번째것...네번째...그 자료의 마지막 의 수 까지 비교해야 한다. 차근차근 인덱스를 증가시켜 하면될 것 같으므로 본인은 for문을 사용할 것이다.
```

```
    //for문
```

```
    초기값:맨앞의 수가 될것이다. 그렇다면 여기서 맨앞의 수의 다음 수 부터 비교할 것이다. =i+1 근데 j가 위에 코드에선 두번째를 가리키니 `j`를 이용하면 될 것 같다!
```

조건:어디까지 비교해야할까? 당연히 마지막의 수까지 돌려야 하므로 배열의 끝까지 비교하면 될 것 같다. 하지만 배열의끝의 인덱스는 -1이므로(인덱스는 1아니라 0부터 시작해서) <=이 아니라 <을 써서 -1을 표현해주면 될 것 같다.(물론 -1을해줘도 될 것같죠?)

증감:하나하나씩 이니깐 ++로해주면될 것같다.

```
    //이제 포문을 만들어보자
```

```
    //포문의 {}범위는 위에 두번째까지비교하는 명령어를 반복하는 것이므로 위에 비교명령어코드를 감싸주어야한다
```

```
        NSInteger j=i+1;
```

```
        for(j;j<unSortedArray.count;j++){//여기서부터
```

```
            NSInteger remainderValue = [unSortedArray[j] integerValue];
```

```
            //맨앞의 수와 옆에있는 수인 두번째수를 나머지를 가져와 `비교`한다.
```

```
            if(firstValue>remainderValue){
```

```
                //맨앞의수보다 뒤에수가 작으므로
```

```
                //맨앞의 수의 수와 자리를 바꾼다.
```

```
                [unSortedArray exchangeObjectAtIndex:i withObjectAtIndex:j];//자리를 바꾸는 메서드를 사용!
```

//아직 firstValue는 if문 밖에있어서 비교해서 작은수는 remainderValue이므로 이제 수를 바꿔준다.

```
        firstValue=remainderValue;
```

```
    }
```

```
    }//여기까지
```

```
    return unSortedArray;
```

```
}
```

이렇게하면 하나하나씩이 끝났다.

나머지 그다음 맨앞의 수를 진행하자

그다음 맨앞의 수를 정한다는 것은 맨앞의 수가 정렬되었으니 두번째가 맨앞이되어서 두번째를 정렬하라는 것이다.

위에서 두번째는 j였다 . j는 무엇인가?  $i+1$ 이었다. 즉 맨처음 앞의 수의 다음번째를 i에 넣어준다면 두번째가 맨처음 수가 된다. 그렇다면 맨위에 키워드 설명에서 i가 전체배열의 길이의 마지막전까지 반복해서 맨처음 수가 되면 될것같다.

이것도 차례대로 진행하므로 for문을 쓸 것이다.

for문

초기식 : 맨처음부터 해야하므로  $i=0$  이 될것이다.

조건 : 배열의 전체길이의 마지막 전까지는 **하나하나씩**의 조건 과 같다 .(-1과 마지막전은 같은 얘기) 그래서 <배열의 길이로 할것이다.

증감:이것또한 맨처음이 차근차근 바뀌므로 ++로!

코드로 구현해보자 !

```
+(NSArray*)doThisSelectionSort:(NSAraay*)data{
```

```
    NSMutableArray *unSortedArray =[[NSMutableArray alloc] initWithArray:dat  
a];
```

```
    //기존의 맨앞의 수
```

```
        NSInteger i=0;
```

```
    //그다음 맨앞의 수 의 for문을 만들어보자
```

```
    for(i;i<unSortedArray.count;i++){
```

```
        NSInteger firstValue = [unSortedArray[i] integerValue];
```

```
    //하나하나씩
```

```
    NSInteger j=i+1;
```

```
    for(j;j<unSortedArray.count;j++){//여기서부터
```

```
        NSInteger remainderValue = [unSortedArray[j] integerValue];
```

```
        //맨앞의 수와 옆에있는 수인 두번째수를 나머지를 가져와 `비교`한다.
```

```
        if(firstValue>remainderValue){
```

```
            //맨앞의 수의 수와 자리를 바꾼다.
```

```
            [unSortedArray exchangeObjectAtIndex:i withObjectAtIndex:j];//자리를 바  
꾸는 메서드를 사용!
```

```
            firstValue=remainderValue;
```

```
        }
```

```
    }// 하나하나씩 여기까지
```

```
    } //그다음맨처음수는 여기까지 하나하씩을 포함해야하므로
```

```
    return unSortedArray;
```

```
}
```