#SORT

정렬은 데이터를 어떠한 기준에 의해서 일렬로 나열하는 것이다.

우리가 실생활에서도 정렬하는 경우가 굉장히 많다.

컴퓨터또한 여러가지 데이터를 정렬을 해야할 때 우리는 적정한 방법으로 컴퓨터가 수행 할 수있도록 해줘야 할 것이다.

지금부터 기본적인 정렬법 몇가지를 소개하고 간단히 그 쓰임을 알아 볼 수 있도록 하겠다.

1. Selection Sort

한국말로는 선택 정렬이다. 어떠한 자료를 오름차순으로 정렬해야 한다면 컴퓨터에게 하나하나 씩 남은 수를 기존의 맨앞의 수 비교 하여 기존의 맨앞의 수보다 작으면 맨앞의수 와 자리를 바꾼다 여기서 맨앞의수는 정렬된다. 이제 그다음 수가 맨앞이 된다. 다음수에서 또 남은 수 중에서 다음 수보다 작다면 맨앞의 수와 자리를 바꾸게 한다.

이것이 선택정렬이다. 위에 저 한국어를 이제 objective-c로 바꿔서 컴퓨터에게 알려주면 자료가 들어왔을 때 수행 할 수 있을 것이다.

먼저 강조된 부분을 해석해서보자!

기존의 맨앞의 수: 정렬되지않는 데이터가 들어왔을때 처음 수(정렬되있지않으니 가장 작은 수일수도 아닐수도 있다.)

비교: 오름차순으로 정렬한다 했으니 작은 것인지를 비교한다고 예상 할 수 있다.

맨앞의 수의 수와 지리를 바꾼다. : 비교해서 조건이 맞다면 바꿔서 맨앞의 수에 가장 작은수가 오 게한다.

하나하나 씩: 들어온 데이터중에서 맨앞 수와 그다음부터 작은 수 인지 하나하나 나머지 수들을 비교 해야한다.

그 다음수가 맨앞이 된다 . :맨앞의 수가 저기의 과정을 통하면 가장 작은수가 맨앞에오니 그 다음수를 맨앞으로 정해서 또 나머지를 비교해서 넣는다.

그다음 수가 맨앞: 어디까지 맨앞이될까? 바로 데이터를 맨끝의 전의 수까지 맨앞의 수가될 것이다. 왜 나면 마지막 전의 수는 가 비교할 것은 마지막 밖에없는데 비교하면 이제 마지막의수는 그다음마지막 수가 없기때문에!!

Step by Step

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 보
내주는 것이기에 {}안에서 정렬 후 다시 배열형식으로 외부로 보낼 것이다.
+(NSArray*)doThisSelectionSort:(NSAraay*)data{
//들어온 배열들은 대부분 고정된 배열을 쓴다. 우리는 고정된 길이가 아닌 데이터를 다루는데 좀더 편
리하고,여러가지 유용한 메서드를 사용 할 수 있도록 NSMutableArray의 배열형식으로 data를 un
SortedArray에 넣어서 사용 할 것이다.
NSMutableArray *unSotedArray =[[NSMutableArray alloc] initWithArray:dat
a];
//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 = [unSotedArray[i] integerValue];
//비교
//이제 맨앞의 수인 0과 나머지 데이터들을 비교 해야한다.우리는 옆에있는 것(i+1)을 검사 할 것
이다.
   NSInteger j=i+1;
   NSInteger remainerValue = [unSotedArray[j] integerValue];
   //맨앞의 수와 옆에있는 수인 두번째수를 나머지를 가져와 `비교`한다.
   if(firstValue>remainerValue){
   //맨앞의 수의 수와 지리를 바꾼다.
    [unSotedArray exchangeObjectAtIndex:i withObjectAtIndex:j];//자리를 바
꾸는 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.

```
NSMutableArray *unSotedArray =[[NSMutableArray alloc] initWithArray:dat
a];
//기존의 맨앞의 수
   NSInteger i=0;
   NSInteger firstValue = [unSotedArray[i] integerValue];
/ /하나하나씩
// 옆의있는 데이터가 한개아니라 여러개라면 두번째것을 비교하고 그다음 세번째것...네번째..그 자료
의 마지막 의 수 까지 비교해야 한다. 차근차근 인덱스를 증가시켜 하면될 것 같으므로 본인은 for문
을 사용할 것이다.
//fot문
초기값:맨앞의 수가 될것이다. 그렇다면 여기서 맨앞의 수의 다음 수 부터 비교할 것이다. =i+1 근
데 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 감싸
주어야한다
NSInteger j=i+1;
for(j;j<unSortedArray.count;j++){//여기서부터
   NSInteger remainerValue = [unSotedArray[j] integerValue];
   //맨앞의 수와 옆에있는 수인 두번째수를 나머지를 가져와 `비교`한다.
   if(firstValue>remainerValue){
   //맨앞의수보다 뒤에수가 작으므로
   //맨앞의 수의 수와 지리를 바꾼다.
    [unSotedArray exchangeObjectAtIndex:i withObjectAtIndex:j];//자리를 바
꾸는 메서드를 사용!
   //아직 firstValue는 if문 밖에있어서 비교해서 작은수는 remainerValue이므로 이제
수를 바꿔준다.
      firstValue=remainerValue;
   }
} / /여기까지
return unSortedArray;
}
```

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

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+1이였다. 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 *unSotedArray =[[NSMutableArray alloc] initWithArray:dat
a];
//기존의 맨앞의 수
   NSInteger i=0;
//그다음 맨앞의 수 의 for문을 만들어보자
for(i;i<unSortedArray.count;i++){</pre>
   NSInteger firstValue = [unSotedArray[i] integerValue];
//하나하나씩
NSInteger j=i+1;
for(j;j<unSortedArray.count;j++){//여기서부터
   NSInteger remainerValue = [unSotedArray[j] integerValue];
   //맨앞의 수와 옆에있는 수인 두번째수를 나머지를 가져와 `비교`한다.
   if(firstValue>remainerValue){
   //맨앞의 수의 수와 지리를 바꾼다.
    [unSotedArray exchangeObjectAtIndex:i withObjectAtIndex:j];//자리를 바
꾸는 메서드를 사용!
       firstValue=remainerValue;
}// 하나하나씩 여기까지
} //그다음맨처음수는 여기까지 하나하씩을 포함해야하므로
return unSortedArray;
}
```