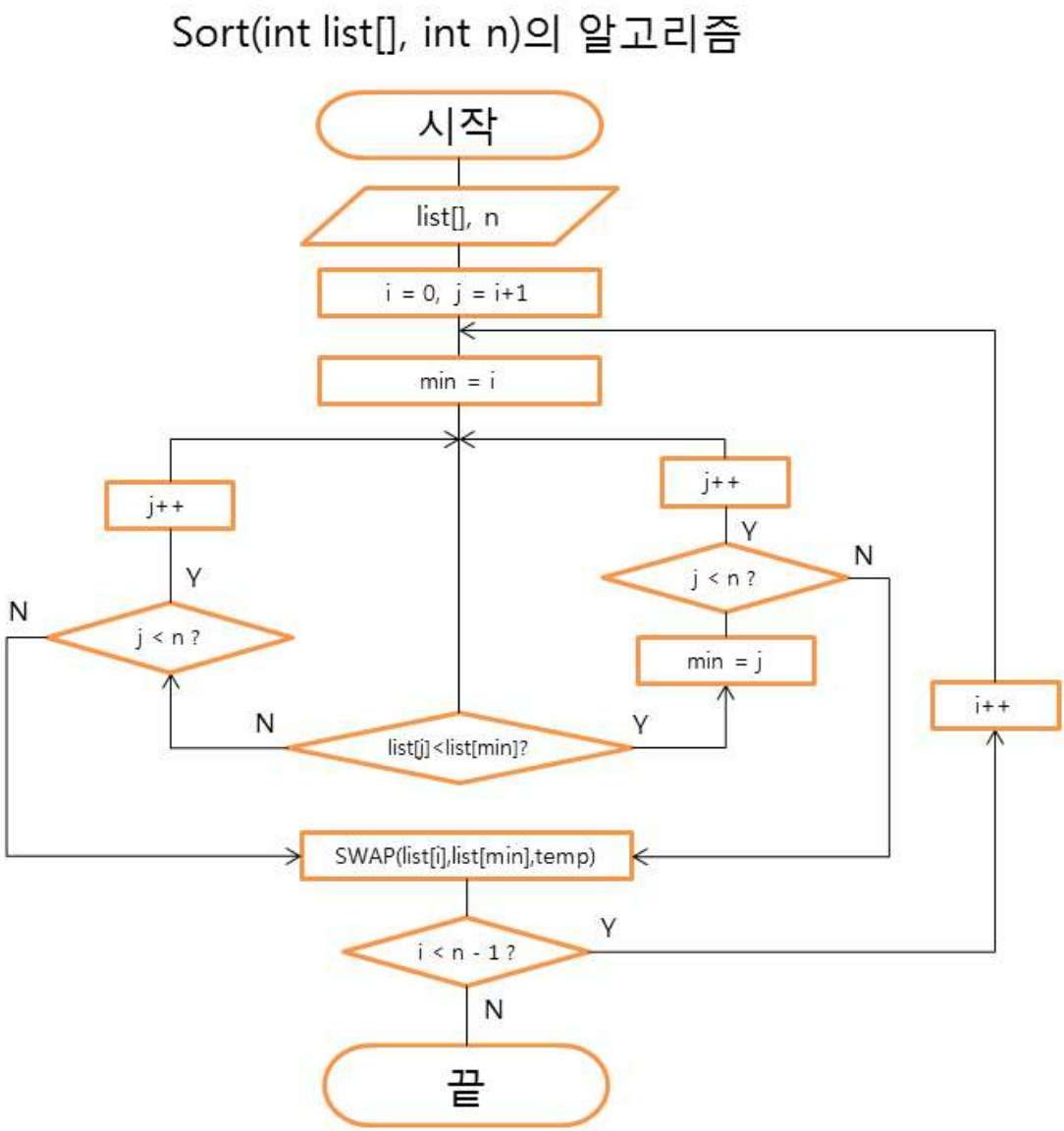
1. 선택정렬

21410785 황 희

1. 개요

- 선택정렬[sort(int list[], int n)]의 알고리즘

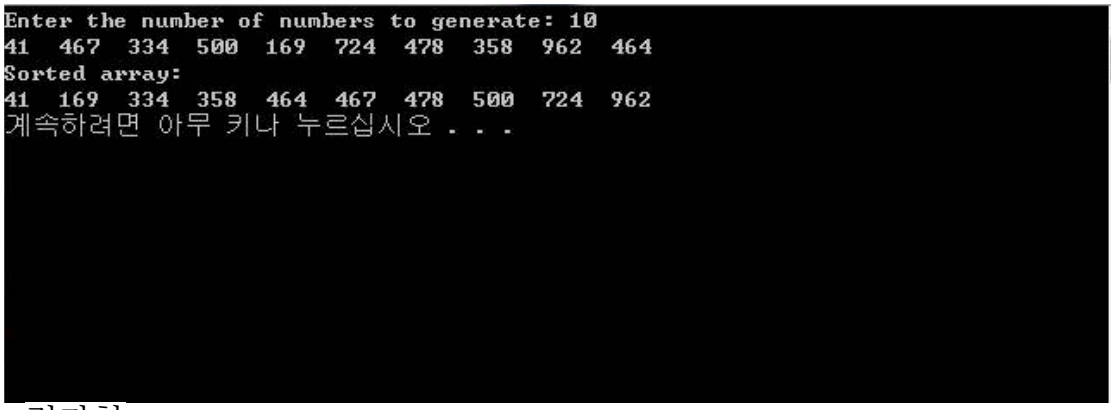


매개변수로 리스트의 주소 list와 리스트 길이 n을 받는다. 리스트의 인덱스를 표시할 변수 i,j를 선언해주고, i는 0, j는 i+1으로 초기화 해준다. 최솟값이 있을 인덱스를 나타낼 변수 min은 우선 i로 설정해준다. 그리고 list[j]와 list[min]을 비교해 list[j]가 더 작다면 min인덱스를 j로 바꿔준다. 이 작업을 j < n 인 동안 반복해주고, n에 도달한다면 미리 지정해둔 SWAP매크로를 통해 최소값이 들어있는 리스트의 위치 list[min]과 i번째 들어있는 요소 list[i]를 교환해준다. 그리고 i를 1더해주고, j를 다시 i+1로 초기화 시켜준다. 위의 과정을 i < n-1인 동안 반복한다.

2. 본문

- 코드(설명은 주석으로 대체)

```
#include <stdio.h> //입출력함수를 사용하기위해 추가함
#include<stdlib.h> //rand()함수를 사용하기위해 추가함
#define MAX_SIZE 101 //MAX_SIZE는 앞으로 101이라는 숫자로 쓰겠다.
#define SWAP(x,y,t) ((t)=(x), (x)=(y), (y)=(t))  //스왑함수를 매크로로 구현
void sort(int list[], int n)
{
	int i, j, min, temp;
	for (i = 0; i<n - 1; i++) {  // list[i]부터 list[n-1]까지 정렬.
		min = i;   // i 위치에 들어갈 최소값의 초기화
		for (j = i + 1; j < n; j++)
			if (list[j] < list[min])
				min = j;// 더 작은 것이 있으면 최소값을 이 곳으로…
		SWAP(list[i], list[min], temp);  // 최소값과 i의 내용을 교체
	}
}
void main(void)
{
	int i, n, list[MAX_SIZE]; // 변수 선언
	printf("Enter the number of numbers to generate: ");
	scanf("%d", &n); // 리스트 길이 입력
	if (n<1 || n>MAX_SIZE) { // 숫자를 잘못 입력했을 경우 종료
		fprintf(stderr, "Improper value of n\n");
		exit(1);
	}
	for (i = 0; i<n; i++) {  // n개의 정수를 random하게 생성
		list[i] = rand() % 1000; // 1000보다 작은 랜덤숫자 생성후 배열에 삽입
		printf("%d  ", list[i]); // 숫자 삽입마다 숫자출력
	}
	sort(list, n);        // sort 함수를 호출. 인자는 배열과 정수의 개수
	printf("\nSorted array:\n");
	for (i = 0; i<n; i++)        // 정렬된 정수를 출력
		printf("%d  ", list[i]);
	printf("\n");
}
```



- 결과창

생성하고자하는 배열 길이를 10으로 설정하자 1~1000사이 10개 숫자를 무작위로 배열에 삽입하였다. 그 다음 sort함수를 이용하여 길이가 10인 배열을 정렬해주었고, 그 결과를 출력하였다.

3. 결론

선택정렬의 원리는 최솟값을 찾아 첫번째 배열에 넣고, 그다음으로 가장 작은 값을 다음 배열에 넣고 하는 반복의 연속이므로 반복에 사용되는 i,j값의 설정을 잘 해주는 것이 중요하다. j는 반드시 i보다 커야한다. 만약 j가 i보다 크지 않다면 이미 작은 값이 앞에 있다면 다시 최솟값의 위치와 i번째의 배열위치를 바꾸게 되므로 선택정렬의 의미가 없다. 그리고, swap함수는 주소에 의한 전달이 되도록 하여야 두 값이 서로 교환이 가능하지만, 여기서는 리스트의 주소를 바로 바꾸므로 굳이 주소에 의한 전달이 필요하지는 않다.

2017-12-03

황 희