

선택정렬 알고리즘

수업: 자료구조

학과: 항공컴퓨터전공

학번: 202200654

이름: 김효준

수강 요일: 월요일

제출일: 2023-03-26

```

#include<stdio.h> //표준 입출력 함수를 사용할 수 있게하는 헤더파일
#include<math.h>
#define MAX_SIZE 101 //배열의 최대 길이
#define SWAP(x,y,t) ((t)=(x), (x)=(y), (y)=(t)) //두 변수의 값을 서로 바꾸는 매크로 함수
void sort(int[], int);

void main(void){
    int i, n; //i: 접근할 element의 index ; n: list의 길이
    int list[MAX_SIZE]; //정렬할 배열

    //배열의 길이를 입력받음
    printf("Enter the number of numbers to generate: ");
    scanf("%d", &n);

    //n이 허용 가능한 범위를 넘어섰을 때 오류처리
    if (n<1 || n>MAX_SIZE){
        fprintf(stderr, "Improper value of n\n");
        exit(1);
    }

    //n개의 element에 각각 랜덤한 값을 삽입
    for(i=0;i<n;i++){
        list[i]=rand() % 1000;
        printf("%d ", list[i]);
    }

    sort(list, n); //리스트 정렬

    //정렬된 리스트를 출력
    printf("\n Sorted array:\n");
    for(i=0;i<n;i++)
        printf("%d ", list[i]);
    printf("\n");
}

//sort함수 정의
void sort(int list[], int n){ //list: 정렬 대상인 배열 ; n: list의 길이
    int i, j, min, temp;
    //i: Access하고 있는 element의 index.
    //j: list[i]와 비교하고 있는 element의 index.
    //min: i번째 작은 수의 index
    //temp: list[i]와 list[j]의 값을 서로 바꾸기 위한 임의의 변수
    for(i=0;i<n-1;i++){
        min=i; //먼저 index가 i인 element를 최소값으로 설정한다.
        //index가 i+1인 element부터 차례로 비교해간다.
        for(j=i+1;j<n;j++){
            //list[j]가 list[min]보다 작으면 min에 j를 대입한다.
            if(list[j]<list[min])
                min=j;
        }
        SWAP(list[i], list[min], temp); //i번째로 작은 값을 가진 element의 값과 index가 i인 element의 값을 서로 바꾼다.
    }
}

```

Enter the number of numbers to generate: 10

41 467 334 500 169 724 478 358 962 464

Sorted array:

41 169 334 358 464 467 478 500 724 962

Process exited after 5.285 seconds with return value 10

계속하려면 아무 키나 누르십시오 . . .

```

#include<stdio.h> //표준 입출력 함수를 사용할 수 있게하는 헤더파일
#include<stdlib.h>
#include<time.h>
#define MAX_SIZE 101 //배열의 최대 길이
#define SWAP(x,y,t) ((t)=(x), (x)=(y), (y)=(t)) //두 변수의 값을 서로 바꾸는 매크로 함수
void sort(int[], int);

void main(void){
    int i, n; //i: 접근할 element의 index ; n: list의 길이
    int list[MAX_SIZE]; //정렬할 배열
    srand((unsigned int)time(NULL)); //rand()함수가 시간에 따라 랜덤한 값을 가지도록 하는 함수.

    //배열의 길이를 입력받음
    printf("Enter the number of numbers to generate: ");
    scanf("%d", &n);

    //n이 허용 가능한 범위를 넘어섰을 때 오류처리
    if(n<1 || n>MAX_SIZE){
        fprintf(stderr, "Improper value of n\n");
        exit(1);
    }

    //n개의 element에 각각 랜덤한 값을 삽입
    for(i=0;i<n;i++){
        list[i]=rand() % 1000;
        printf("%d ", list[i]);
    }

    sort(list, n); //리스트 정렬

    //정렬된 리스트를 출력
    printf("\n Sorted array:\n");
    for(i=0;i<n;i++)
        printf("%d ", list[i]);
    printf("\n");
}

//sort함수 정의
void sort(int list[], int n){ //list: 정렬 대상인 배열 ; n: list의 길이
    int i, j, min, temp;
    //i: Access하고 있는 element의 index.
    //j: list[j]와 비교하고 있는 element의 index.
    //min: i번째 작은 수의 index
    //temp: list[i]와 list[j]의 값을 서로 바꾸기 위한 임의의 변수
    for(i=0;i<n-1;i++){
        min=i; //먼저 index가 i인 element를 최솟값으로 설정한다.
        //index가 i+1인 element부터 차례로 비교해간다.
        for(j=i+1;j<n;j++){
            //list[j]가 list[min]보다 작으면 min에 j를 대입한다.
            if(list[j]<list[min])
                min=j;
        }
        SWAP(list[i], list[min], temp); //i번째로 작은 값을 가진 element의 값과 index가 i인 element의 값을 서로 바꾼다.
    }
}

```

```

Enter the number of numbers to generate: 10
383 287 751 904 188 281 208 695 848 893
Sorted array:
188 208 281 287 383 695 751 848 893 904

```

```

-----
Process exited after 0.9091 seconds with return value 10
계속하려면 아무 키나 누르십시오 . . .

```

```

#include<stdio.h> //표준 입출력 함수를 사용할 수 있게하는 헤더파일
#include<stdlib.h>
#include<time.h>
#define MAX_SIZE 101 //배열의 최대 길이

void sort(int[], int);
void swap(int* n1, int* n2);

void main(void){
    int i, n; //i: 접근할 element의 index ; n: list의 길이
    int list[MAX_SIZE]; //정렬할 배열
    srand((unsigned int)time(NULL)); //rand()함수가 시간에 따라 랜덤한 값을 가지도록 하는 함수.

    //배열의 길이를 입력받음
    printf("Enter the number of numbers to generate: ");
    scanf("%d", &n);

    //n이 허용 가능한 범위를 넘어섰을 때 오류처리
    if(n<1 || n>MAX_SIZE){
        fprintf(stderr, "Improper value of n\n");
        exit(1);
    }

    //n개의 element에 각각 랜덤한 값을 삽입
    for(i=0; i<n; i++){
        list[i]=rand() % 1000;
        printf("%d ", list[i]);
    }

    sort(list, n); //리스트 정렬

    //정렬된 리스트를 출력
    printf("\n Sorted array:\n");
    for(i=0; i<n; i++)
        printf("%d ", list[i]);
    printf("\n");
}

//매크로 함수인 SWAP을 일반 함수로 바꾼 swap
void swap(int * n1, int * n2){ //두 변수의 주소를 입력받음
    int tmp = *n1;
    *n1=*n2;
    *n2=tmp;
}

//sort함수 정의
void sort(int list[], int n){ //list: 정렬 대상인 배열 ; n: list의 길이
    int i, j, min;
    //i: Access하고 있는 element의 index.
    //j: list[i]와 비교하고 있는 element의 index.
    //min: i번째 작은 수의 index

    for(i=0; i<n-1; i++){
        min=i; //먼저 index가 i인 element를 최소값으로 설정한다.
        //index가 i+1인 element부터 차례로 비교해간다.
        for(j=i+1; j<n; j++){
            //list[j]가 list[min]보다 작으면 min에 j를 대입한다.
            if(list[j]<list[min])
                min=j;
        }
        swap(&list[i], &list[min]); //i번째로 작은 값을 가진 element의 값과 index가 i인 element의 값을 서로 바꾼다.
    }
}

```

```

Enter the number of numbers to generate: 10
871 160 93 193 650 31 357 345 531 697
Sorted array:
31 93 160 193 345 357 531 650 697 871

-----
Process exited after 8.082 seconds with return value 10
계속하려면 아무 키나 누르십시오 . . .

```