Kruskal’ s algorithm using C

2019320139 Choi Yun Ji

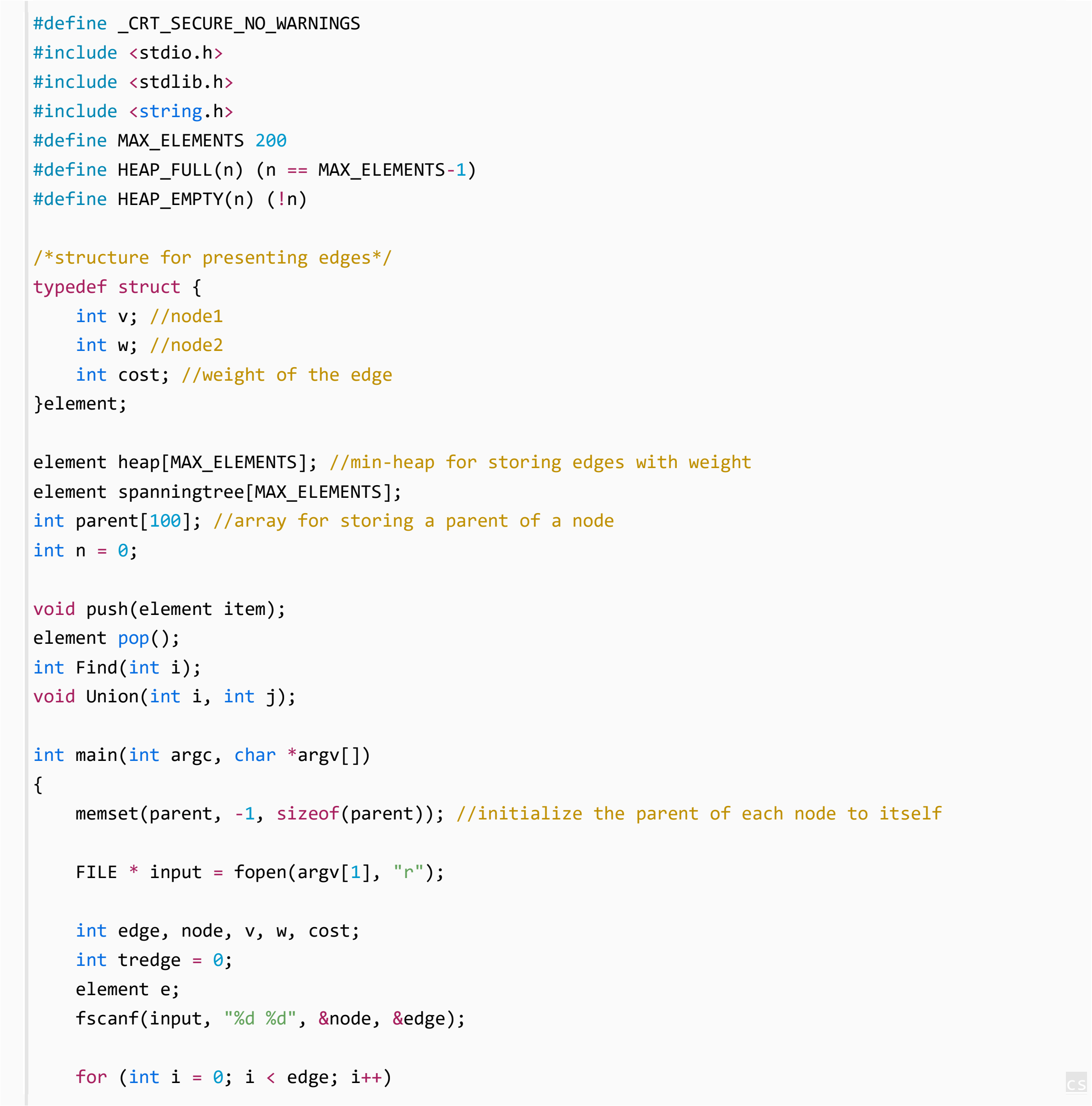
1. Development environment

OS : Window 10

IDE : Visual Studio 2017

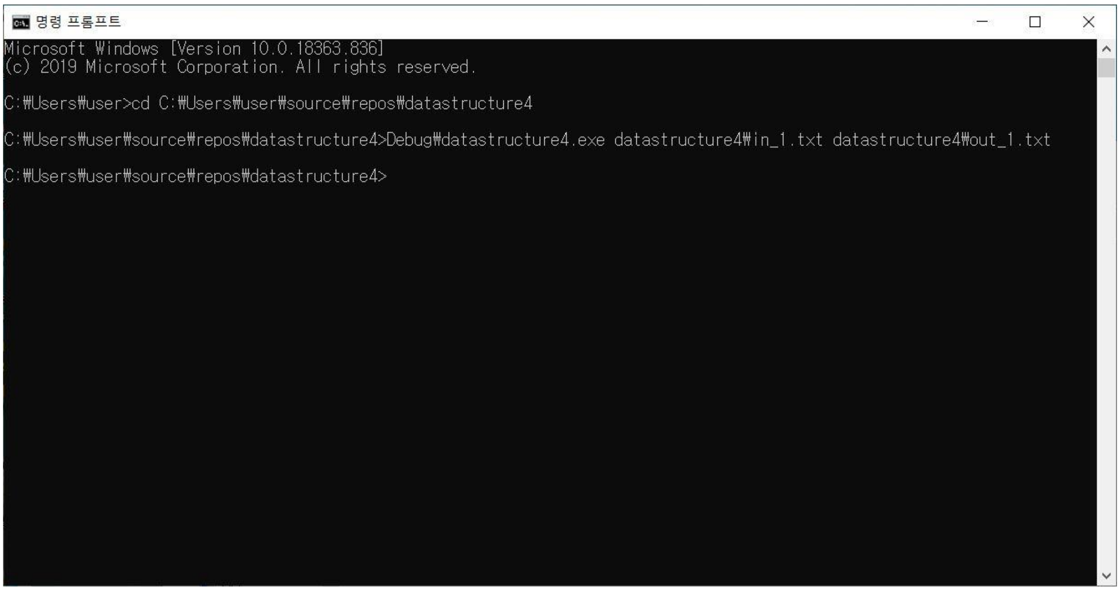
2. Explanation of the algorithm and the code

The explanation was replaced by annotations.

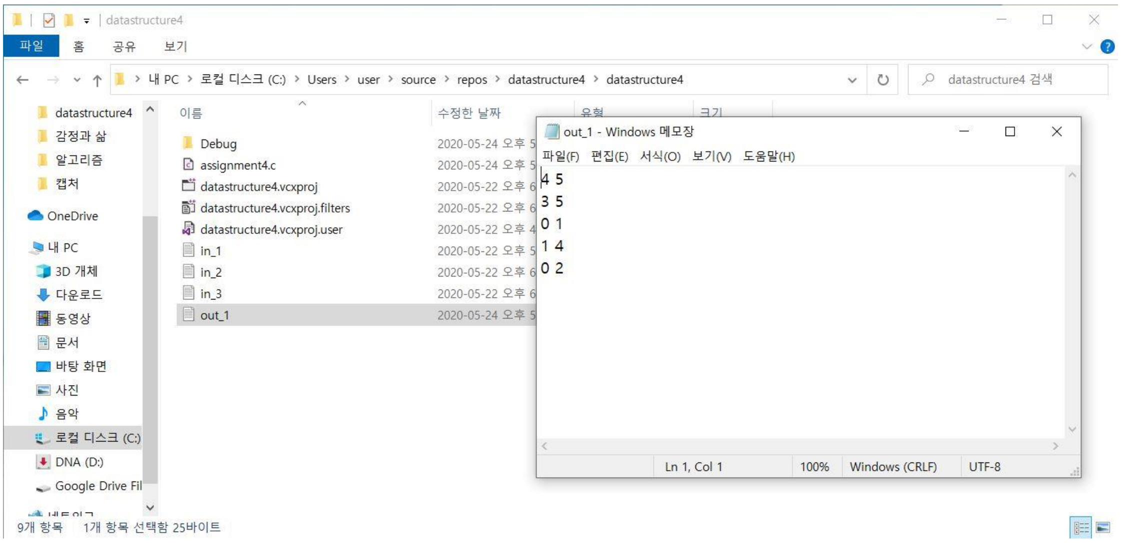


```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_ELEMENTS 200
#define HEAP_FULL(n) (n == MAX_ELEMENTS-1)
#define HEAP_EMPTY(n) (!n)

/*structure for presenting edges*/
typedef struct {
    int v; //node1
    int w; //node2
    int cost; //weight of the edge
}element;

element heap[MAX_ELEMENTS]; //min-heap for storing edges with weight
element spanningtree[MAX_ELEMENTS];
int parent[100]; //array for storing a parent of a node
int n = 0;

void push(element item);
element pop();
int Find(int i);
void Union(int i, int j);

int main(int argc, char *argv[])
{
    memset(parent, -1, sizeof(parent)); //initialize the parent of each node to itself

    FILE * input = fopen(argv[1], "r");

    int edge, node, v, w, cost;
    int tredge = 0;
    element e;
    fscanf(input, "%d %d", &node, &edge);

    for (int i = 0; i < edge; i++)
    {
        fscanf(input,"%d %d %d", &v, &w, &cost);
        element item = { v, w, cost }; //make new edge
        push(item); //push the item to min-heap
    }
    fclose(input);
    int f1, f2;

    while (tredge < node - 1 && !HEAP_EMPTY(n))
        //Tree contains less than n-1 edges && heap is not empty
    {
        e = pop();
        if ((f1 = Find(e.v)) != (f2 = Find(e.w)))
            //if parent of e.v and parent of e.w are not same -> (e.v, e.w) doesn't create a cycle
        {
            spanningtree[tredge++] = e; //add (e.v, e.w) to Tree
            Union(f1, f2); //combine a tree with f1 as root and a tree with f2 as root
        }
    }

    if (tredge != node - 1) //if tree contains fewer than n-1 edges
    {
        printf("No spanning tree...\n");
        return 0; //exit the program
    }

    FILE * output = fopen(argv[2], "w");

    for (int i = 0; i < tredge; i++)
    {
        fprintf(output, "%d %d\n", spanningtree[i].v, spanningtree[i].w);
        //print edges of minimum cost spanning tree
    }

    fclose(output);
}
```

3. Result of execution



```
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:₩Users₩user>cd C:₩Users₩user₩source₩repos₩datastructure4

C:₩Users₩user₩source₩repos₩datastructure4>Debug₩datastructure4.exe datastructure4₩in_1.txt datastructure4₩out_1.txt

C:₩Users₩user₩source₩repos₩datastructure4>
```



이름	수정한 날짜
Debug	2020-05-24 오후 5
assignment4.c	2020-05-24 오후 5
datastructure4.vcxproj	2020-05-22 오후 6
datastructure4.vcxproj.filters	2020-05-22 오후 6
datastructure4.vcxproj.user	2020-05-22 오후 4
in_1	2020-05-22 오후 5
in_2	2020-05-22 오후 6
in_3	2020-05-22 오후 6
out_1	2020-05-24 오후 5

```
4 5
3 5
0 1
1 4
0 2
```