1. 코드 분석 및 주석

#include <stdio.h>

#include <stdlib.h>

#define TRUE 1

#define FALSE 0

#define MAX\_VERTICES 8

#define INF 1000L

int weight[MAX\_VERTICES][MAX\_VERTICES] =

{ {0,3,INF,INF,INF,INF,INF,12},

{3,0,8,INF,INF,INF,INF,10},

{INF,8,0,15,2,INF,INF,INF},

{INF,INF,15,0,INF,INF,INF,INF},

{INF,INF,2,INF,0,9,4,5},

{INF,INF,INF,INF,9,0,INF,INF},

{INF,INF,INF,INF,4,INF,0,6},

{14,10,INF,INF,5,INF,6,0} };

struct distance {

int key;//정렬 후 순서

int dist;

}dist\_st[MAX\_VERTICES];

int found[MAX\_VERTICES];

void build\_min\_heap(void) {

int i = 1;

int length = MAX\_VERTICES;

int child, parent;

int temp;

while (i != length / 2) {

parent = i;

child = i \* 2;

if (dist\_st[child - 1].dist > dist\_st[child + 1 - 1].dist)//두 개의 자식 노드 중 더 작은 애 선택

child = child + 1;

while ((dist\_st[parent - 1].dist > dist\_st[child - 1].dist)) {

//key의 정렬 순서를 바꿔준다.

temp = dist\_st[parent - 1].key;//배열 올바르게 만들기 위해서 각각 -1 해준다.

dist\_st[parent - 1].key = dist\_st[child - 1].key;

dist\_st[child - 1].key = temp;

parent = child;

if (child > MAX\_VERTICES)

break;

child = parent \* 2;

if (dist\_st[child - 1].dist > dist\_st[child + 1 - 1].dist)//두 개의 자식 노드 중 더 작은 애 선택

child = child + 1;

}

i++;

}

}

int Extract\_Min() {//정렬값이 0인 걸 return 한다.

int min;

int i;

for (i = 0; i < MAX\_VERTICES; i++) {

if (found[i] == FALSE) {//queue에서 제거되지 않고 가장 작은 최소의 수

min = i;

break;

}

}

return min;

}

void Decrease\_key\_min\_heap(int i, int key) {

dist\_st[i].dist = key;

build\_min\_heap;

}

void shortest\_path(int start, int n) {

int i, u, w;

for (i = 0; i < n; i++) {

dist\_st[i].dist = weight[start][i];

found[i] = FALSE;

}

build\_min\_heap();

found[start] = TRUE;

dist\_st[start].dist = 0;

for (i = 0; i < n - 2; i++) {

u = Extract\_Min;

found[u] = TRUE;

for (w = 0; w < n; w++) {

if (!found[w]) {

if (dist\_st[u].dist + weight[u][w] < dist\_st[w].dist) {

Decrease\_key\_min\_heap(w, dist\_st[u].dist + weight[u][w]);

}

}

}

}

}

int main() {

shortest\_path(0, MAX\_VERTICES);

}

1. 결과 및 콘솔 창

이것 역시도 출력이 불가능했다.

1. 문제점.

이것도 2번 문제와 같이 내 지식 수준에서는 문제점을 발견하지 못했다. Dist가 새로 업데이트 될 때 마다 정렬을 해주는 decrese min heap을 사용했고 또한 정렬된 순서와 vertices가 통합되지 않는 것을 염려하여 heap node를 선언하고 그 안에 vertice 값을 저장해서 코드를 짰지만 역시 출력이 되지 않았다. 어디서부터 출력이 막히는지 알아보기 위해서 일일히 printf를 해보았는데 Extract\_min 이 부분에서부터 출력이 막히는 것 같아. 하지만 그 이유는 잘 모르겠다.

내 추측으로 보건데 inf 부분이 비교가 안되는 것이 아닐까 생각해본다. 코드를 작성하기 위해서 몇 날 며칠을 매달렸지만 결국 풀지 못해 너무 아쉽다.