ITP 20002-03 Discrete Mathematics, Fall 2021

Homework 4

Kim Hyun Uk

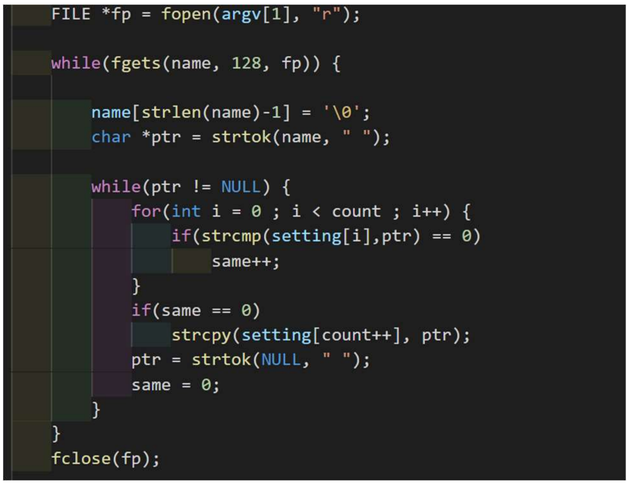
21800201@handong.edu

1. Introduction

Hasse Diagram is actually catch the point of undirected graph. Before I print the output of the input data, I have to summarize the traits of indegree. The Input data is formed as “left (space) right” formation. Saving each Name as a number and the array which have the information of indegree and level should be ready. Also, when there is an self to self direction in input file, I have to continue the next line.

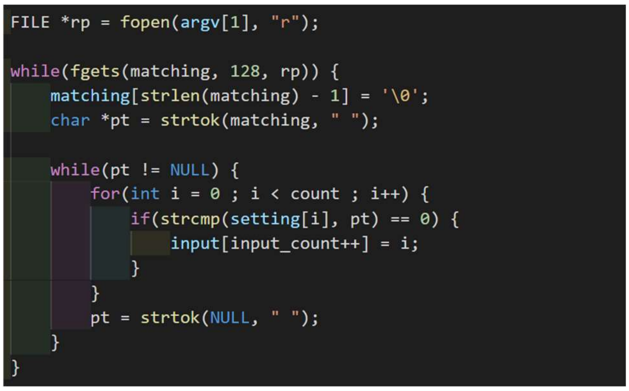
Actually, it should be solved using Topological Algorithm, But It needs queue or stack concept. In C language, there are no libraries for using that concept easily. So I approached this with using appropriate array. It’s much easier to use and solve the problem.

2. Approach



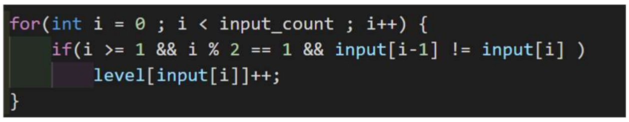
```
FILE *fp = fopen(argv[1], "r");

while(fgets(name, 128, fp)) {

    name[strlen(name)-1] = '\0';
    char *ptr = strtok(name, " ");

    while(ptr != NULL) {
        for(int i = 0 ; i < count ; i++) {
            if(strcmp(setting[i],ptr) == 0)
                same++;
        }
        if(same == 0)
            strcpy(setting[count++], ptr);
        ptr = strtok(NULL, " ");
        same = 0;
    }
}
fclose(fp);
```

This code for read the file and save all input data without redundancy. Setting array will save all data of Input file.



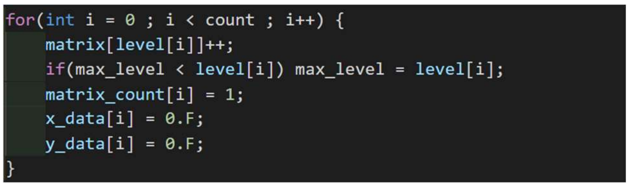
```
FILE *rp = fopen(argv[1], "r");

while(fgets(matching, 128, rp)) {
    matching[strlen(matching) - 1] = '\0';
    char *pt = strtok(matching, " ");

    while(pt != NULL) {
        for(int i = 0 ; i < count ; i++) {
            if(strcmp(setting[i], pt) == 0) {
                input[input_count++] = i;
            }
        }
        pt = strtok(NULL, " ");
    }
}
```

This code is for conversion of Input file. The “Input1.dat” file is filled with English but it is too difficult to save and use in another situation. Therefore, I changed each word to number in array name “input”.



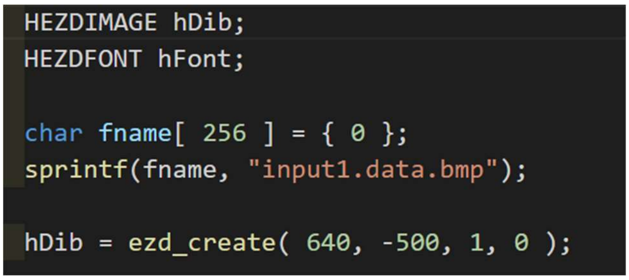
```
for(int i = 0 ; i < input_count ; i++) {
    if(i >= 1 && i % 2 == 1 && input[i-1] != input[i] )
        level[input[i]]++;
}
```

This code is for saving the indegree data of each node in array name “level”. Each node has an information of indegree as to Input data file. First, I have to consider the odd index of input array. Because odd number is secondary course of lecture and it is part of indegree counting. And input[i-1] != input[i] means that I have to avoid the situation of self to self direction. In this Input, the Java -> Java.



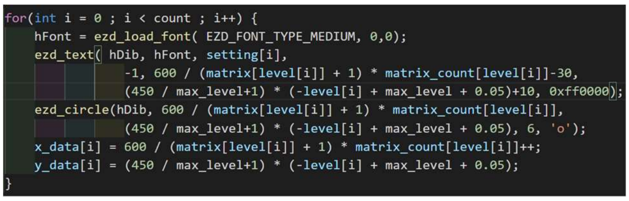
```
for(int i = 0 ; i < count ; i++) {
    matrix[level[i]]++;
    if(max_level < level[i]) max_level = level[i];
    matrix_count[i] = 1;
    x_data[i] = 0.F;
    y_data[i] = 0.F;
}
```

This code is for Drawing. “matrix” array is saving the number of each level. “max_level” variable saves the maximum number of indegree. “x_data and y_data” array save each coordinate of vertex.



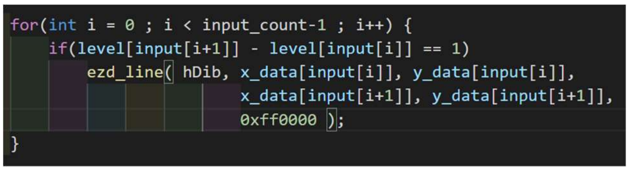
```
HEZDIMAGE hDib;
HEZDFONT hFont;

char fname[ 256 ] = { 0 };
sprintf(fname, "input1.data.bmp");

hDib = ezd_create( 640, -500, 1, 0 );
```

Setting the size of screen.



```
for(int i = 0 ; i < count ; i++) {
    hFont = ezd_load_font( EZD_FONT_TYPE_MEDIUM, 0,0);
    ezd_text( hDib, hFont, setting[i],
                -1, 600 / (matrix[level[i]] + 1) * matrix_count[level[i]]-30,
                (450 / max_level+1) * (-level[i] + max_level + 0.05)+10, 0xff0000);
    ezd_circle(hDib, 600 / (matrix[level[i]] + 1) * matrix_count[level[i]],
                (450 / max_level+1) * (-level[i] + max_level + 0.05), 6, 'o');
    x_data[i] = 600 / (matrix[level[i]] + 1) * matrix_count[level[i]]++;
    y_data[i] = (450 / max_level+1) * (-level[i] + max_level + 0.05);
}
```

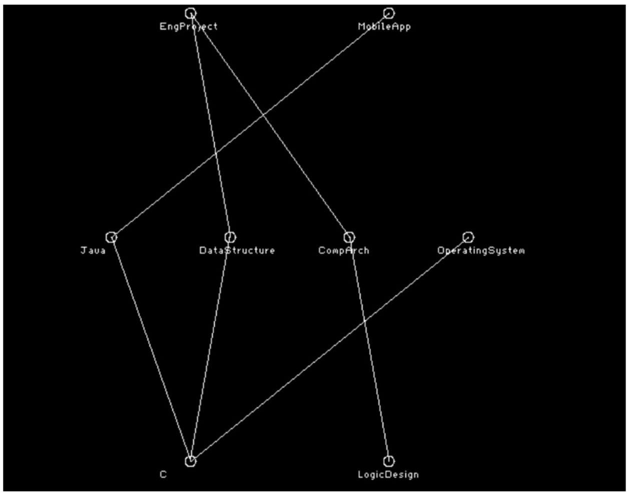
Set the text and vertex(circle) and Save the coordinates of each vertex to x_data and y_data. To make the graph more beautiful, Width and Height gap should have the rule. I have to count the vertex of each floor. With that, divide the width using that information. Likewise, I have to count the max indegree of Input data file.



```
for(int i = 0 ; i < input_count-1 ; i++) {
    if(level[input[i+1]] - level[input[i]] == 1)
        ezd_line( hDib, x_data[input[i]], y_data[input[i]],
                        x_data[input[i+1]], y_data[input[i+1]],
                        0xff0000 );
}
```

This code is for drawing line of graph. Using for loop, when the gap of the level for each input line is 1, draw the line with 2 coordinates.

3. Evaluation



4. Discussion

Before I implemented this code, I studied Topological Algorithm. Its main idea is using queue. But C language don't have the library of this. And the preparation for this was a little bit taking some time to implementation. Therefore I saved all data with Array.

I was wonder how to reduce the number of crossed line. In my case, All data was save as the sequence of Input data file. Therefore I didn't implement the rotation of that. In my test case, It should be changed Java and Operating System.

5. Conclusion

I learned that Hasse Diagram summarize the data more readable. This makes us to comprehend the intension of maker. I totally understand the concept of Graph during solving this problem. Thanks for your teaching.