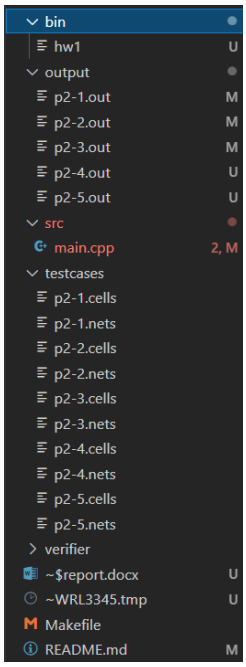
CENG5270 Hw1: Two-way Min-cut Partitioning

Ziyi Wang 1155168629

File structure

The file structure of my program is shown as below:



Compile and Execute

To compile the project, you should follow the steps below:

In the root directory, type the command: 'make'. It will generate the executable file "hw1" in the directory './bin'.

If you want to remove it please type the command: 'make clean' in the root directory,

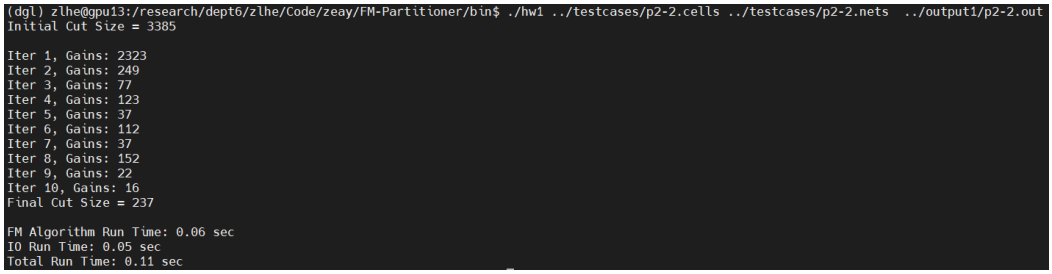
To execute the project, you should follow the steps below:

1. In the root directory, type the command: 'cd ./bin' to go to the bin directory.
2. In the bin directory, type the command: /<exe> <cells file name> <nets file name> <output file name>

   e.g. ./hw1 ../testcases/p2-1.cells ../testcases/p2-1.nets ../outputs/p2-

1.out

An execution example is shown as below:



Result

The results are shown as follows:

Testcase	Cut Size	T_IO	T_FM	T_total
1	6	0.001	0.004	0.005
2	237	0.04	0.05	0.09
3	922	0.62	2.04	2.66
4	45945	1.34	10.16	11.5
5	125277	4.47	3.04	7.51

Detail of Implementation

1. Difference from slide

My implemented algorithm is exactly the same described in class . I also implement a my version of bucket list data structure which is almost the same as that described in the slide. The only difference might be that each time when I insert a cell into the structure, I just insert it into the front of the corresponding linked list instead of keeping the corresponding linked list sorted by cell size. This will reduce runtime while causing some performance drop.

2. How to restore

Firstly, I use a variable k to record the times of cell movement (k is incremented when a cell is moved.) I also use some variables to record the accumulated gain as well as the maximum accumulated gain and its corresponding best k. When one pass of the FM algorithm is over, I can get the maximum partial sum from the maximum accumulated gain. And I use a vector to store the order of cell movement, so when the whole algorithm is over, I can restore the result through moving the first best k cells in the vector.

Analysis

1. Comparison with previous top 5 results

I do not achieve better results but the difference is not that big. One reason might be that my version of bucket list is not sorted by the size. It means that each time when I select a cell with maximum gain, it might not be of the smallest size and violation of balance is more likely to happen. A solution might be sort the bucket list by the size.

2. What I have learned

Firstly, I'm more familiar with the FM algorithm and can fully understand it now. It is true that we can learn better from practice. Meanwhile, I did encounter some problems in this homework, like misunderstanding of the algorithm, some segement faults, etc. Solving these problems help me capture a clearer picture of FM algorithm.