

Assignment 2

Objective:

Develop Python code for recursively partitioning of a given set of blocks till each partition has a single block using FM algorithm

Given: Number of blocks N , Dimension of each of the blocks (width, height), Adjacency matrix which is a $N \times N$ matrix. Each element in the adjacency matrix (i, j) gives the number of interconnects running between them.

The inputs are given in a file named input_1.txt and input_2.txt. The code developed by you should read these files, process it and provide the expected output. For testing your code I will have my own test case with a larger 'N' and the corresponding data placed in file named input_1.txt and input_2.txt. Your code should read the file and produce the output.

You may use the relation $pmax = dmax \cdot wmax$, where $dmax = \max$ degree of a vertex (# edges incident to it) and $wmax$ is the maximum edge weight (Area of the block) for computing the weight.

Print appropriate messages along with the outputs. Comment the code appropriately. There is a weightage for the comments, messages printed during execution and the graphical depiction of the output.

Expected Outputs:

1) MUST draw the groups of blocks as partitioned enclosed hierarchically. Each group when partitioned must use the same colour within the group and different colour for different level of hierarchy. Print the corresponding block number within the every block. Eg: (1, 2) are grouped together in a partition, then put black outlined blocks 1 and 2 enclosed in a single red outlined block. Similarly (3, 4) are grouped in a partition together, then put black outlined blocks 3 and 4 enclosed in a single red outlined block. The two red outlined boxes together in a single green coloured box and so on.

2) Print the total weight each partition.

Optional additional output: Try to plot the partition tree using the Graphviz tool (Dotty which is available freely as open source tool in linux). If you can generate (create or write) the partition.dotty text file, your efforts will be appreciated.

Prove that your code works with the following test case: The test data can be found in two files namely input_1.txt and input_2.txt.

The data in input_1.txt provides the sizes of the blocks and can be read in the following format: {6, {10, 5}, {8, 6}, {4, 13}, {10, 3}, {8, 4}, {4, 1}} where the first data 6 indicates the number of blocks, then each of the six pairs of data enclosed in the flower braces specify {width, height} of the individual block.

The data in input_2.txt defines the connectivity between the blocks and can be read as 6X6 adjacency matrix which gives the number of edges between the blocks.

```
{0 1 3 2 4 2
1 0 3 1 2 1
3 3 0 1 1 2
2 1 1 0 2 3
4 2 1 2 0 1
2 1 2 3 1 0}
```