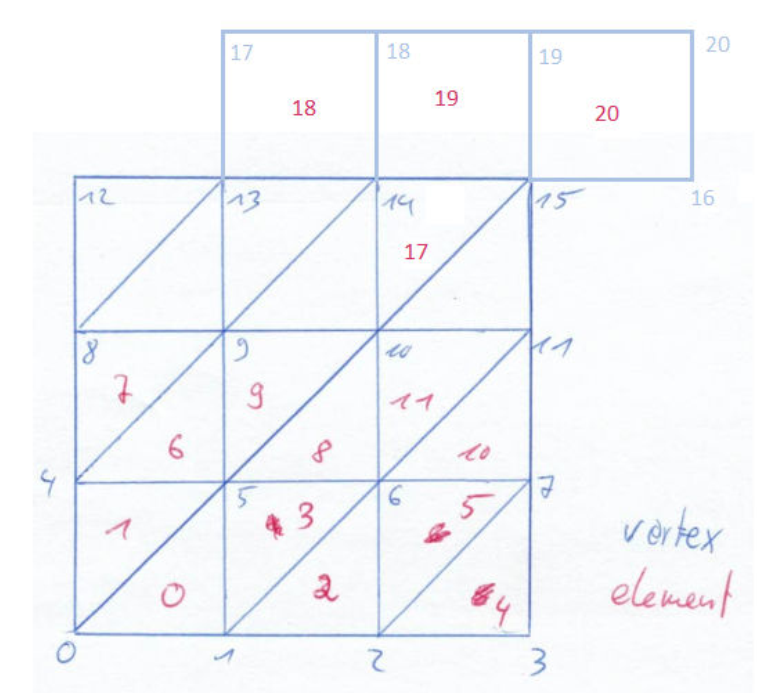
Answers to tasks #1 and #2

How to use. In makefile and regarding to check\_programming.pdf, there are 3 compilers (and their flags). To compile with each compiler the other two compilers should be commented. We named outputs as outgpp, outclang and outicpc that you can rename them!

Task #1.

We implemented task #1 such that unstructured grid (mix of triangles and rectangles) is also acceptable. An example is presented (files end with \_unstructured)



First we read Element and Coordinate files and might print them.

1.a. We construct (and can print) 3 data structures for

- Nodes->Elements : N2C stores for each node, index of cells that share it.

* Elements->Elements : CellNeighb stores for each cell, index of cells that are its neighbor.
* Node->Node : NodeNeighb stores for each node, index of nodes that are its neighbor. Note that two neighboring nodes are not necessarily connected by a edge. For example in a rectangle, diagonal nodes are neighbor with no connecting edge.

1.b. Then coordinates of a point will be asked to find the minimum distance. If the coordinate belongs to a cell, index of the cell will be printed (with zero distance). Otherwise (exterior point), cell index with the nearest Euclidean distance will be printed. It continues by asking another coordinate unless you choose No.

1.c. No change is required.

Task #2

We construct an object of class Graph with the Node -> Node structure from task #1 and can print it.

2.a. Then we derive a weighted (inverse of squared distances) graph (class WGraph) and can print it. Further we assumed diagonal entries (an edge with the same endpoints) with 1e3 as weight;

2.b. A CRS matrix is a weighted graph with matrix entries as weights. So the constructor of class CRSMat, only needs a weighted graph that we derived from WGraph. Moreover, we implemented another constructor to directly employ connectivity and nodes coordinates.

2.c. CRS matrix can be sent to matlab with the output text file : CRS2Matlab.txt that should be loaded in matlab with readFromCpp.m :

*function A = readFromCpp()*

*A = load('CRS2Matlab.txt');*

*A = spconvert(A);*

*end*

So result of matrix-vector product can be easily checked in matlab. Assuming n\_row as number of rows of the matrix, we check CRS matrix-vector product by two vectors: first, all entries are one and second, [1;2;3;…;n\_rows]. Results will be saved in text files.

Finally we read a matrix from matlab. Use write2cpp.m script to write a matrix in matlab to the text file Matlab2CRS.txt. There is a Matlab2CRS example which belongs to A40x40.mat. Copy the text file into current folder of c++ codes (wherever you found CRS2Matlab.txt) and press a key. CRS matrix will be constructed and results of the two abovementioned matrix-vector production will be saved in text file.