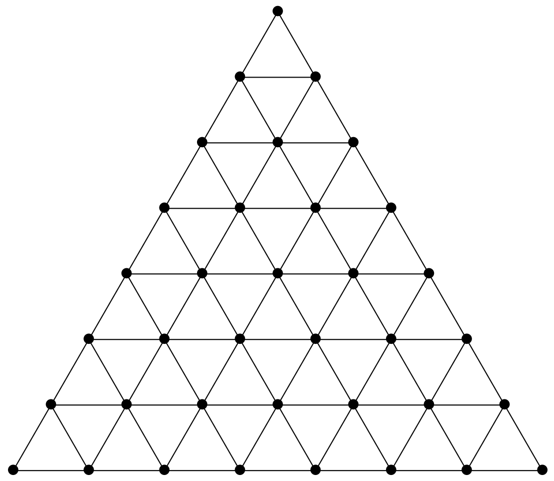
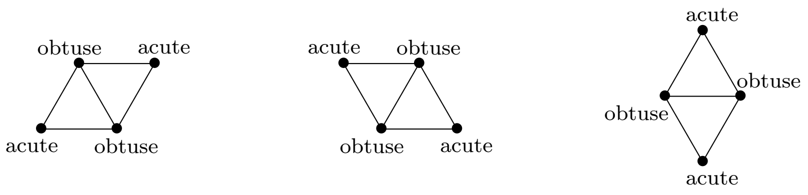
Programming Assignment
Due date: Thursday, December 10th, 2020, noon, via SAM
(`https://sam.csc.liv.ac.uk/COMP/Submissions.pl?strModule=COMP331` or
`https://sam.csc.liv.ac.uk/COMP/Submissions.pl?strModule=COMP557`)

This assignment is worth 10% of your grade.

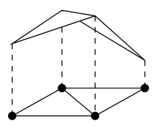
We are given a triangular array of 36 nodes in the plane that are connected among each other as follows.



A *small triangle* is a set of three adjacent nodes. If two small triangles share two nodes, then they form a *small rhombus*. In each small rhombus, two nodes are at *obtuse* corners and two nodes are at *acute* corners:

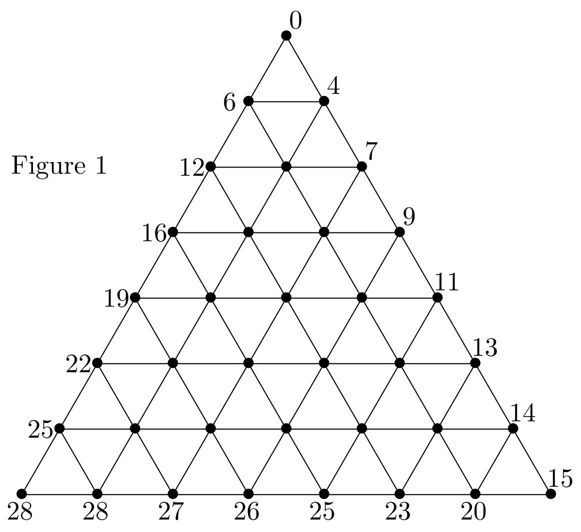


Each of the 36 nodes gets assigned a *height*, which is a real number. Hence, by linear interpolation between the nodes, a height vector $h \in \mathbb{R}^{36}$ gives rise to a triangulation of a surface in 3-dimensional space that on a small rhombus looks as the picture on the right.



We will only consider those height vectors that have fixed heights as follows on the nodes at the boundary:

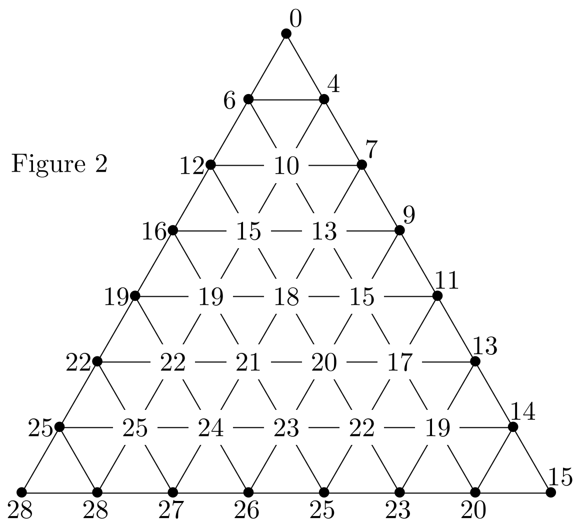
Figure 1



A height vector $h \in \mathbb{R}^{36}$ is called *mountain* if for all small rhombi (in all three rotations) we have that the sum of the heights of the nodes at the obtuse corners is at least as high as the sum of the nodes at the acute corners.

Here is an example of a mountain:

Figure 2



The inequalities for all small rhombi are satisfied, for example $6+4 \geq 10+0$ and $10+4 \geq 6+7$ and $10+6 \geq 12+4$.

A *weighting* $w \in \mathbb{R}^{36}$ is a vector of real numbers. For a given weighting w, the w-weight $\text{weight}_w(h)$ of a mountain h is defined as

$$\text{weight}_w(h) = \sum_{\text{node } v} w(v) \cdot h(v).$$

For a given weighting w, a mountain h is called *w-optimal* if there is no other mountain with a higher w-weight.

The set of mountains whose heights at the boundary nodes are as illustrated in Figure 1 forms a polytope in $\mathbb{R}^{15}$. Your taks is to experimentally determine the number of the polytope's vertices. This shall be done by choosing random linear objective functions and maximising them within the polytope.

1. Write a function (in any programming language) that picks a random weighting w and writes to the disk an `.lp` file whose optimal solution is a w-optimal mountain.
2. Write a program that calls the above function 1000 times to generate 1000 `.lp` files, calls `glpsol` on those files and analyses the output. After this analysis, the program should output the set of distinct mountains that were found.
3. For each mountain that was found, draw a picture analogous to Figure 2. Put them in a single `.pdf` file.

Submission format: Put the following into a single `.zip` file:

1. The program code.
2. The 1000 `.lp` files.
3. The `.pdf` file containing the figures of the mountains analogous to Figure 2.