CAD practical – programming part 7:
Partitioning a polygon into monotone pieces

The zip file "CADCG_W7_Code.zip" available on the web page, contains the code needed for generating the executable. To compile this executable, change the Build Target to "`poly_decomp-static`". Since the algorithm has not been completely written, if you run the executable now it will show a segmentation fault.

Polygon Partitioning

Polygon partitioning with respect to a given line aims to ease the triangulation of complex planner polygons (Theory: ppt 7 pp.24-36).

1. Consider the oriented polygon depicted in the figure on the next page.

   Vertices are indicated v_i and edges e_i.

   To avoid ambiguities, the lexicographic order of the vertices is the following:

   $v_{14} < v_0 < v_{15} < v_{11} < v_1 < v_3 < v_2 < v_{16} < v_{13} < v_{10} < v_{12} < v_4 < v_9 < v_5 < v_8 < v_7 < v_6$

   Using the algorithm described in the slides, decompose firstly the polygon into monotone polygons **by hand** (no coding for the moment!).

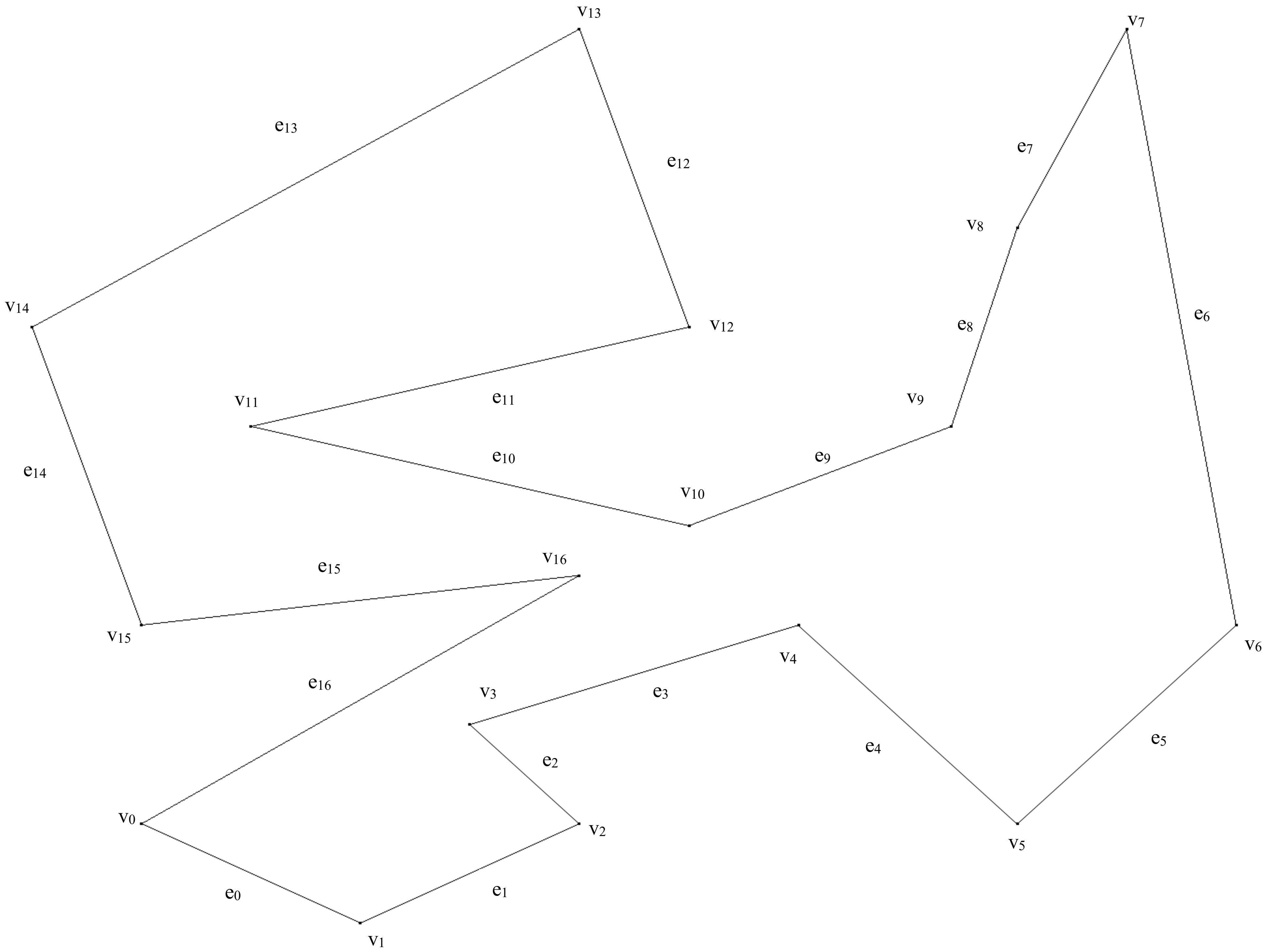
   To verify your work, you are asked to **draw the diagonals** that cut the polygon as predicted by the algorithm, and to provide the **status state after treatment of vertex 3 and the corresponding vertices to the edges contained in the status**.

2. In the file "`test/polygon_decomposition/main.cc`", implement the following function:

```
void handle_regular_vertex(vertex &v, std::set<edge *, Compare> &T,
std::vector<edge> &polygon)
```

This function manages the case for a regular vertex. Its pseudo-code is given in the presentation. Tip: get inspired by the other "handle vertex" functions.

At the end, build and run to check if you obtain the same diagonals as the ones obtained by applying the algorithm by hand (point 1).


v_{13}
v_7
e_{13}
e_{12}
e_7
v_8
v_{14}
v_{12}
e_8
e_6
v_{11}
e_{11}
v_9
e_{14}
e_{10}
e_9
v_{10}
v_{16}
e_{15}
v_{15}
v_4
v_6
e_{16}
v_3
e_3
e_4
e_5
e_2
v_0
v_2
v_5
e_0
e_1
v_1