# Fixing Singular Vertices

How to remove a singular vertex by cutting through the
singularity

1) Allocate new vertex index

2) Copy new vertex coordinates
   From original vertex

3) Fix coordIndex array corner values

# A Simple Example
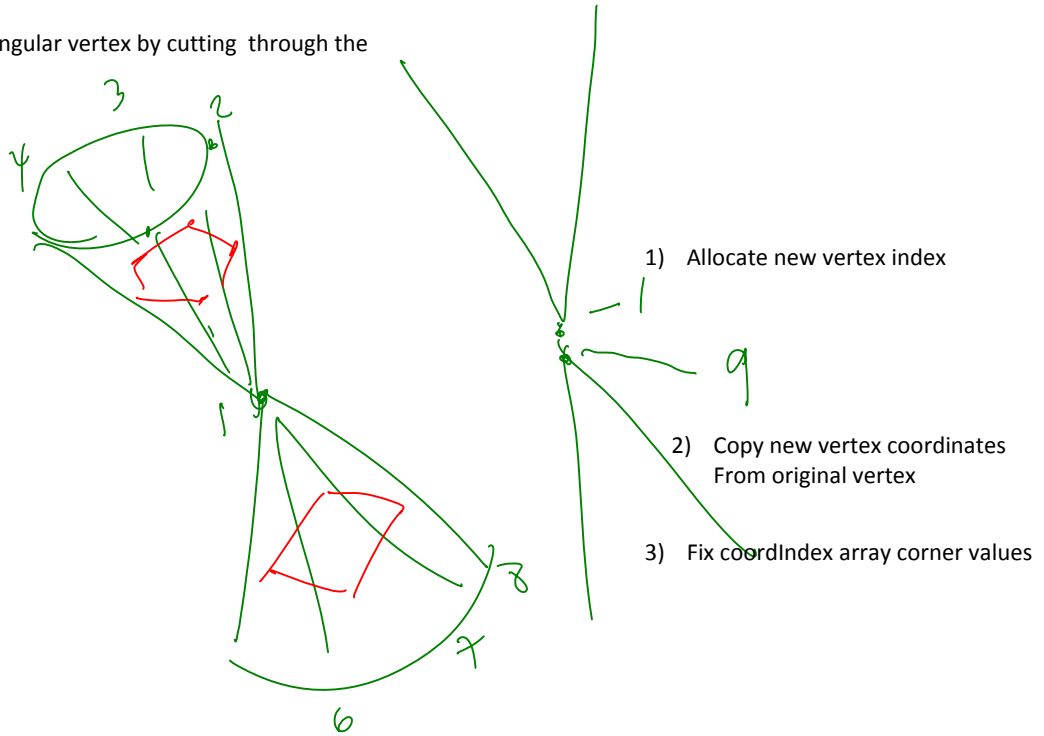
# Cutting through marked edges
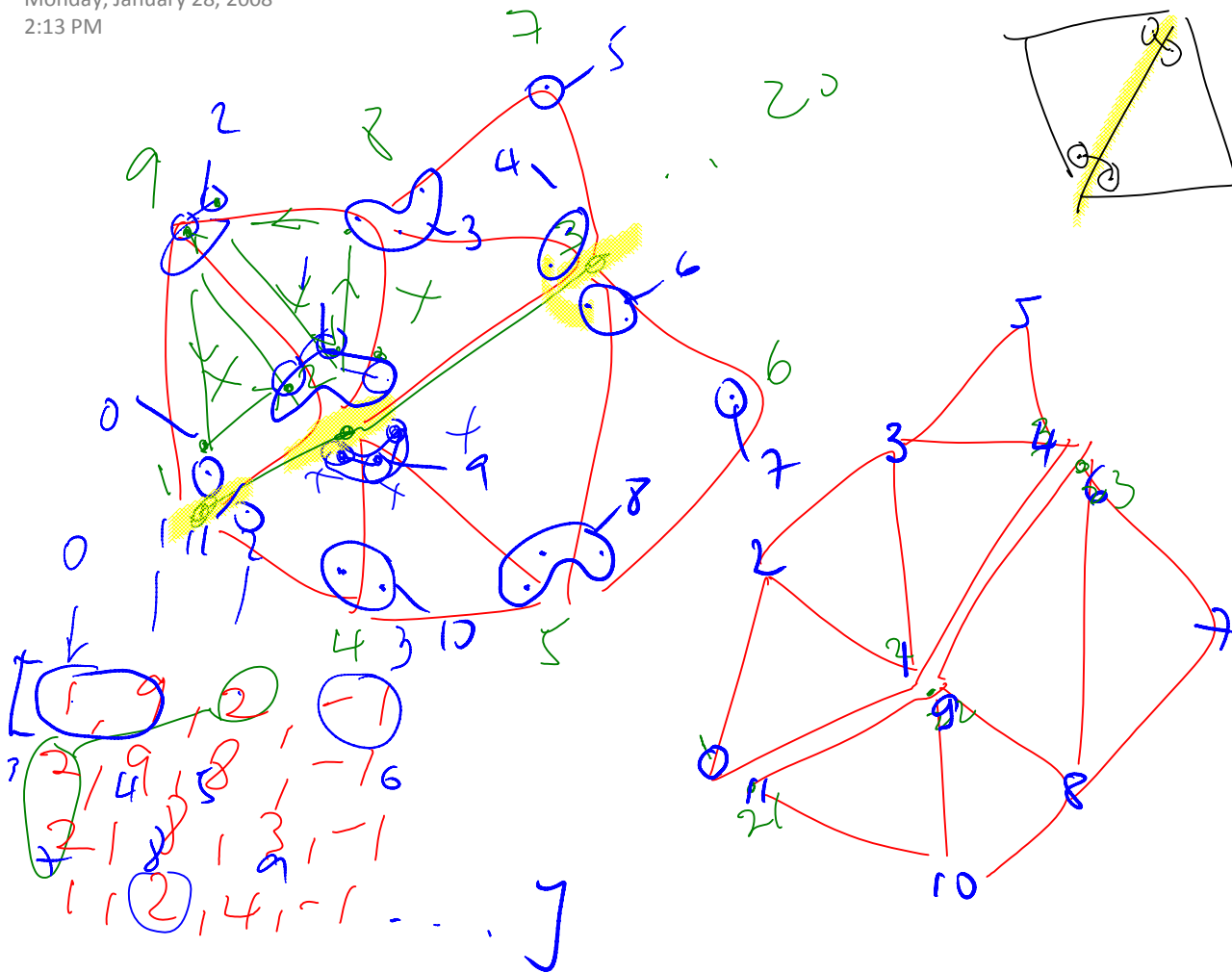
Monday, January 28, 2008
2:13 PM

1) Corners are indices of elements of the coordIndex array
2) We need a data structure to maintain a partition of the corners
3) For each regular edge, not marked as a cut, join the two pairs of twin corners
4) Enumerate the subsets of resulting partition
5) The partition indices are the output vertex indices
6) While traversing the input coordIndex array, write on the output coordIndex array:
for each corner, the partition index it belongs to; keep the -1's
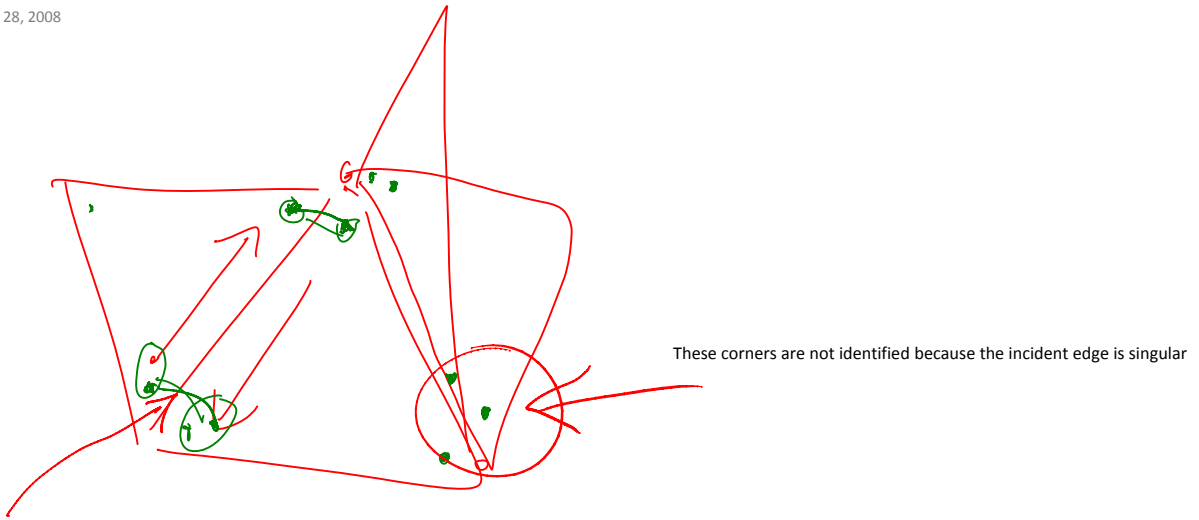
# Partition data structure
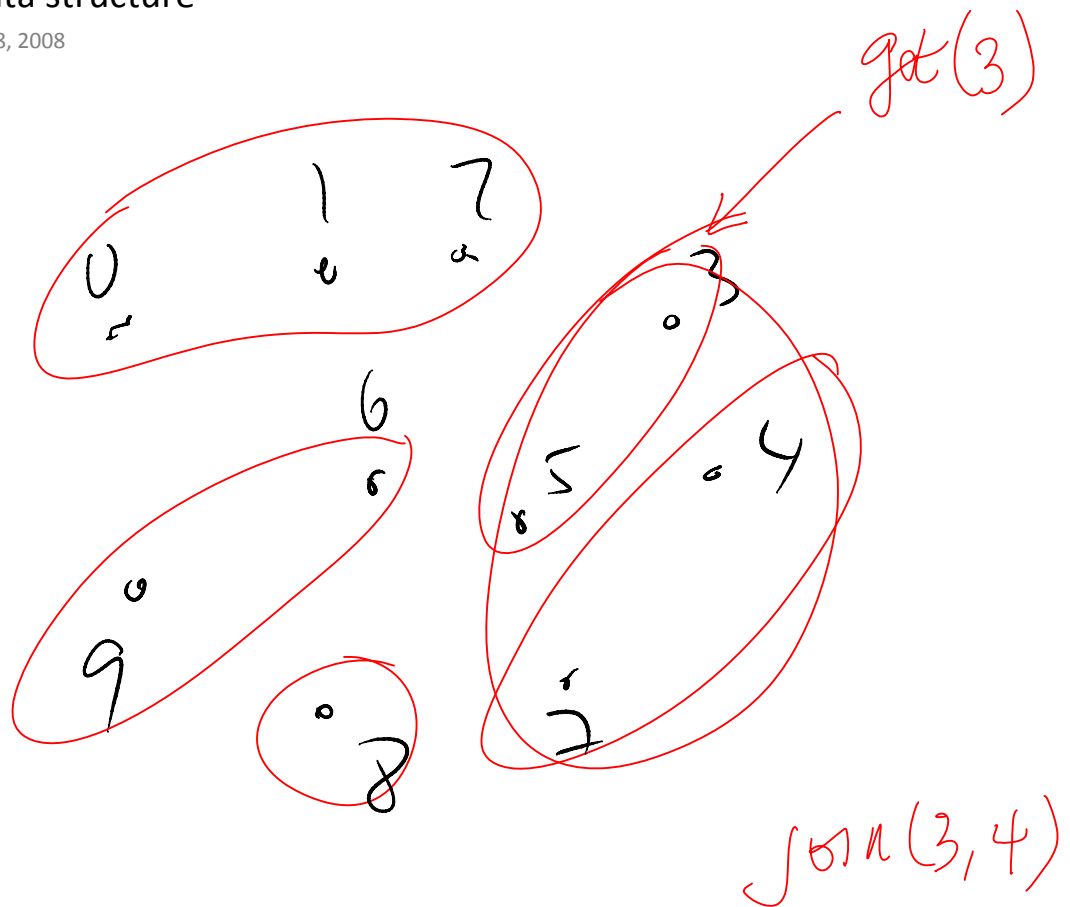
1) Efficient implementation based on Tarjan's Union-Find data structure
2) Maintains a partition ot N numbers {0,…,N-1}
3) Implements two operations: get(i) and join(i,j)
4) get(i) does not return a partition number, but an element of the partition subset I belongs to chosen as the representative for the subset, so that I and j belong to the same subset if and only if get(i)==get(j)
5) After a number of these operations have been applied, the remaining partitions can be enumerated

# Cutting Through Sigular Edges

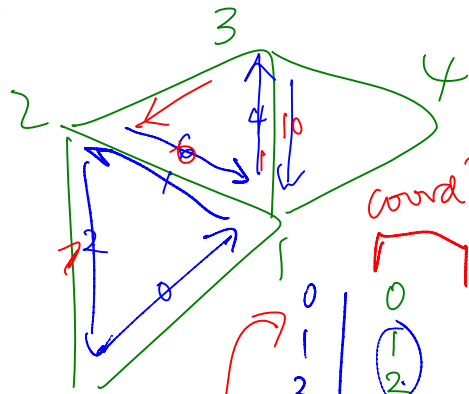These corners are not identified because the incident edge is singular

These two corners are identified because they are opposite to each other through a regular edge
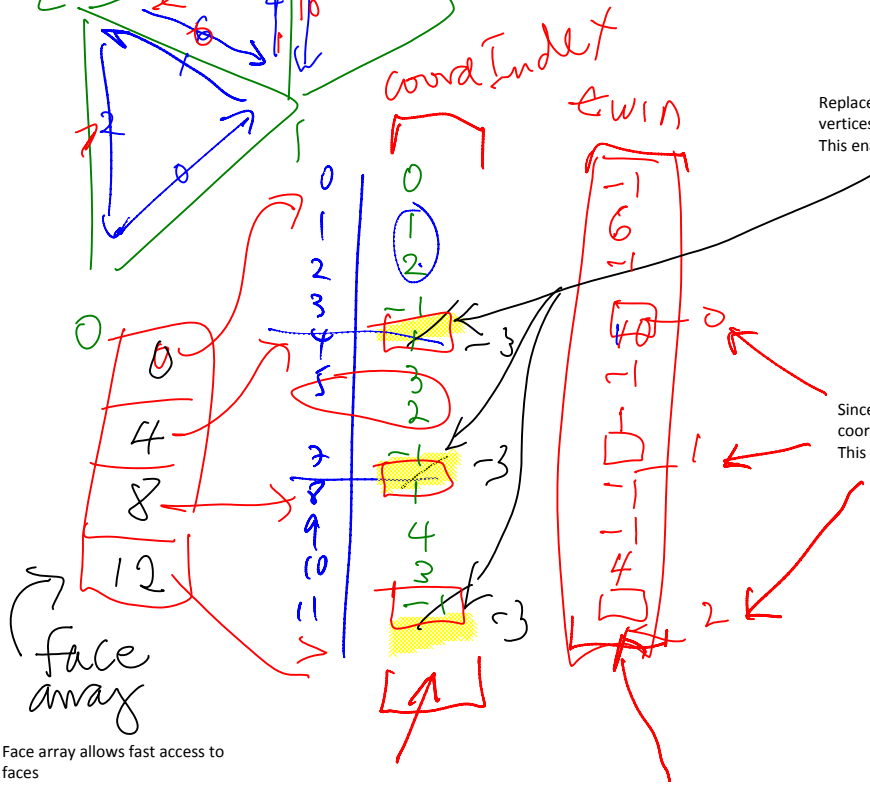
Singular vertices are automatically fixed

# The StaticHalfEdge class

Efficient implementation of HalfEdge data structure

Replace the -1's in coordIndex with -**n** where **n** is the number of vertices of the current face
This enables fast cyclical traversal of vertices within each face.

Since the end-of-face markers are already stored in the coordIndex array, store face indices here
This allows for fast mapping from corner to face index

Face array allows fast access to faces

Original coordIndex array

New array of same length as coordIndex array is used to store the twin edges as corner indices
-1's indicate boundary edges