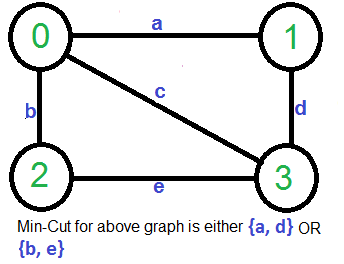
**Karger’s algorithm for Minimum Cut | Set 1 (Introduction and Implementation)**

Given an undirected and unweighted graph, find the smallest cut (smallest number of edges that disconnects the graph into two components).  
The input graph may have parallel edges.

For example consider the following example, the smallest cut has 2 edges.  
[](http://www.geeksforgeeks.org/wp-content/uploads/Kargerfirst.png)

A Simple Solution use [Max-Flow based](http://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/) [s-t cut algorithm](http://www.geeksforgeeks.org/minimum-cut-in-a-directed-graph/) to find minimum cut. Consider every pair of vertices as source ‘s’ and sink ‘t’, and call minimum s-t cut algorithm to find the s-t cut. Return minimum of all s-t cuts. Best possible time complexity of this algorithm is O(V5) for a graph. [How? there are total possible V2 pairs and s-t cut algorithm for one pair takes O(V\*E) time and E = O(V2)].

Below is simple Karger’s Algorithm for this purpose. Below Karger’s algorithm can be implemented in O(E) = O(V2) time.

1) Initialize contracted graph CG as copy of original graph

2) While there are more than 2 vertices.

a) Pick a random edge (u, v) in the contracted graph.

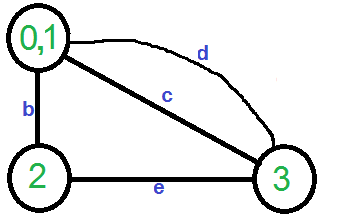
b) Merge (or contract) u and v into a single vertex (update

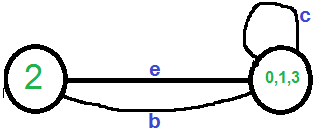
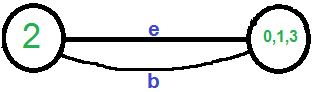
the contracted graph).

c) Remove self-loops

3) Return cut represented by two vertices.

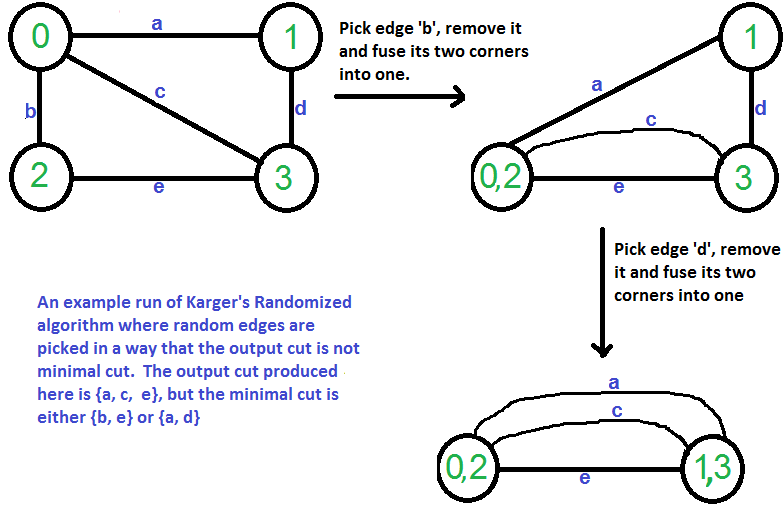
Let us understand above algorithm through the example given.

Let the first randomly picked vertex be ‘**a**‘ which connects vertices 0 and 1. We remove this edge and contract the graph (combine vertices 0 and 1). We get the following graph.  
[](http://www.geeksforgeeks.org/wp-content/uploads/Karger21.png)

Let the next randomly picked edge be ‘d’. We remove this edge and combine vertices (0,1) and 3.  
[](http://www.geeksforgeeks.org/wp-content/uploads/Karger3.png)  
We need to remove self-loops in the graph. So we remove edge ‘c’  
[](http://www.geeksforgeeks.org/wp-content/uploads/Karger4.png)

Now graph has two vertices, so we stop. The number of edges in the resultant graph is the cut produced by Karger’s algorithm.

***Karger’s algorithm is a*** [***Monte Carlo algorithm***](http://en.wikipedia.org/wiki/Monte_Carlo_algorithm) ***and cut produced by it may not be minimum.***For example, the following diagram shows that a different order of picking random edges produces a min-cut of size 3.

[](http://www.geeksforgeeks.org/wp-content/uploads/Karger11.png)