CE-514 : Advanced Algorithms

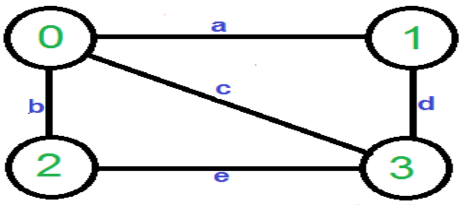
Lab 4: Finding the Minimum Graph Cut using Randomized Algorithm

---

1. **Aim :** Write a program to find the Minimum Graph Cut (Karger's algorithm) using Monte-Carlo type of Randomized algorithm.

2. **Objective** : Students will be able to implement the Monte-Carlo type of Randomized algorithm to find the Graph Cut with a minimum number of cross-edges. Although, it may not give the Minimum Cut but the probability of that can be reduced by running the algorithm $100n^2$ times.

3. Description: Given an undirected and unweighted graph, find the smallest cut i.e. the smallest number of edges that disconnects the graph into two components. The input graph may have parallel edges. For example, consider the below example, the smallest cut has 2 edges.



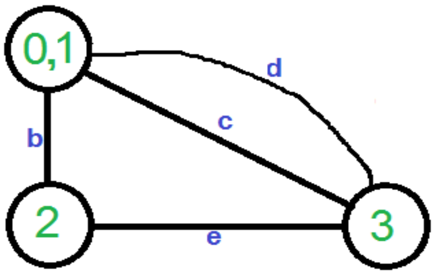
Min-Cut for above graph is either {a, d} OR {b, e}

4. Algorithm and Example:

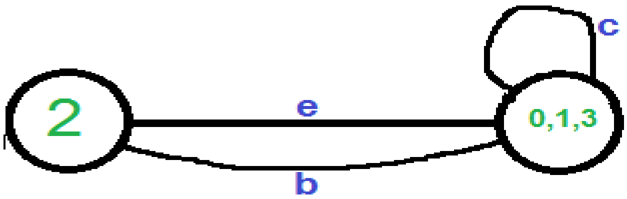
Below is a simple Karger's Algorithm for this purpose. It can be implemented in $O(E) = O(V^2)$ time.

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.

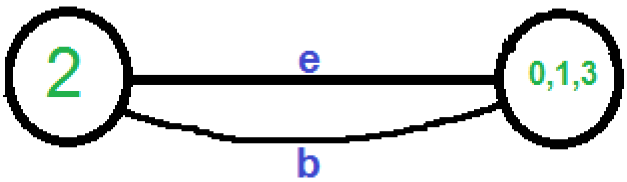
Let us understand the above algorithm through the example given. Let the first randomly picked edge be ‘a‘ which connects vertices 0 and 1. We remove this edge and contract the graph (combine vertices 0 and 1). We get the following graph.



Let the next randomly picked edge be ‘d’. We remove this edge and combine vertices (0,1) and 3.

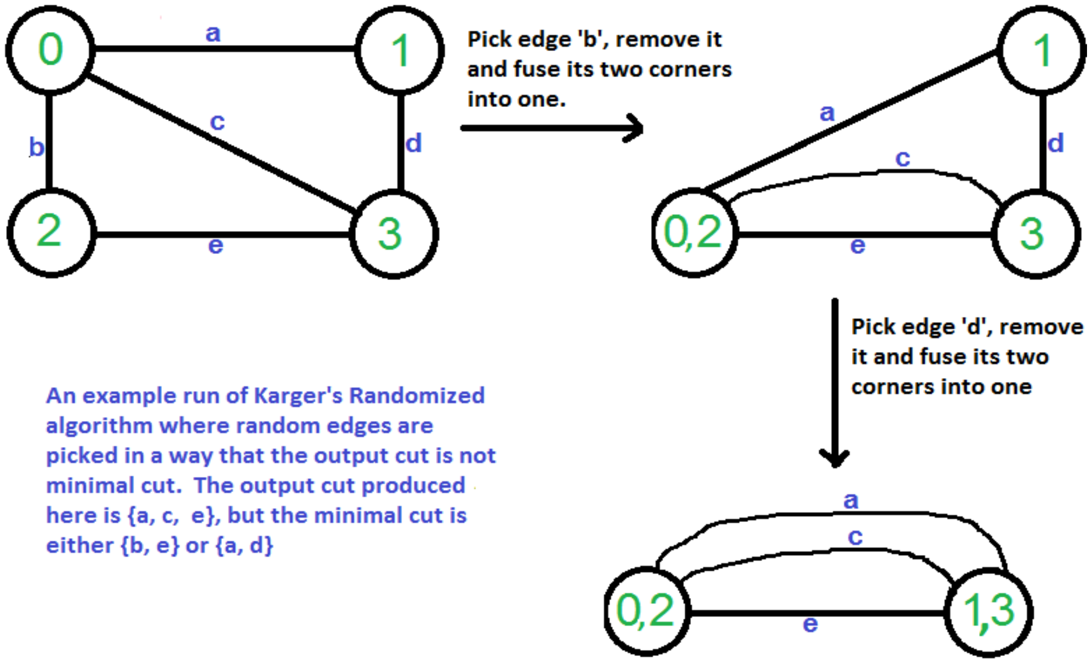


We need to remove self-loops in the graph. So we remove the edge ‘c’.



Now the 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 and the 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. Note that the above program is based on the outcome of a random function and may produce different output.

Here, we have discussed simple Karger’s algorithm and have seen that the algorithm doesn’t always produce min-cut. **The above algorithm produces min-cut with probability greater or equal to that $1/(n^2)$**

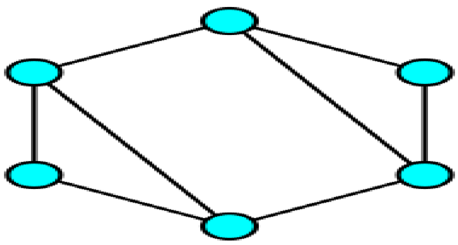


5. Implementation Notes:

- Adjacency Matrix can be used to represent the weighted graph. Contraction of edges can be implemented over the Matrix itself.
- The input graph can also be represented as a collection of edges and **union-find data structure** can be used to keep track of components.

6. Exercise:

1. Consider the following figure and run the randomized Min-Cut algorithm.



2. Justify: Karger's algorithm is the Monte-Carlo algorithm with suitable examples.
3. Derive the probability of error for the above algorithm. How can you make that nearer to 0? Justify your answer in detail.

7. References:

- Introduction to Algorithms Third Edition by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
- COMPUTER ALGORITHMS by Ellis Horowitz University of Southern California Sartaj Sahni University ofFlorida Sanguthevar Rajasekaran University of Florida
- Web Resource: www.geeksforgeeks.org