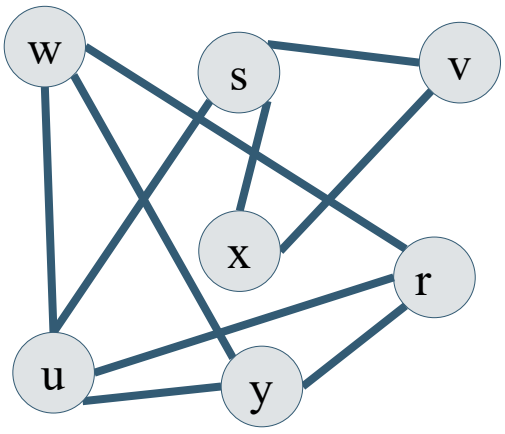
Pseudocode

```
minCut = infinity

int FindMinCut(Edge[1,...,e], Vertex[1,...,v])
   while ( // there are more than two vertices
      edge -> Choose edge randomly from the list
      ContractEdge(edge)
   Return number of edges between the vertices

void ContractEdge(Edge e) // e.u = one vertex of the edge, e.v = second
vertex
      Create new vertex: SuperNode
      Reattach all edges from e.u and e.v to SuperNode
      Delete e
```

Run Karger's Algorithm yourself: Run-Through

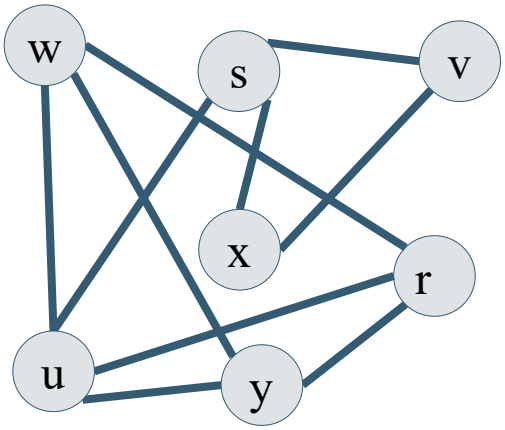


Edges:

- 1 - sv
- 2- vx
- 3- sx
- 4 - su
- 5 - uw
- 6 - wy
- 7 - yr
- 8 - rw
- 9 - uy
- 10 - ur

Random Sequence of Numbers:

Run Karger's Algorithm yourself: Run-Through



Edges:

- 1 - sv
- 2- vx
- 3- sx
- 4 - su
- 5 - uw
- 6 - wy
- 7 - yr
- 8 - rw
- 9 - uy
- 10 - ur

Random Sequence of Numbers:

Probability that output isn't min-cut

- **Fact 1**: If there are n vertices, the expected value of a vertex's degree is 2|E|/n
- **Fact 2**: Size of min-cut is upper-bounded by 2|E|/n
- **Fact 3**: A randomly picked edge crosses the min cut with probability at most 2/n
  - Proof: **Work on this with the people around you**