



# Advanced Algorithms

## Assignment 3: Minimum cut

June 13, 2022

<b>Budai Matteo</b>	2057217
<b>Burke Jamie</b>	2044062
<b>Tlepin Sanjar</b>	2041606

---

# Contents

<b>1</b>	<b>Introduction . . . . .</b>	<b>2</b>
<b>2</b>	<b>Stoer and Wagner's Deterministic Algorithm . . . . .</b>	<b>3</b>
2.1	Data Structure . . . . .	3
2.2	Implementation . . . . .	3
2.3	Complexity . . . . .	3
<b>3</b>	<b>Karger and Stein's Randomized Algorithm . . . . .</b>	<b>4</b>
3.1	Data Structure . . . . .	5
3.2	Implementation . . . . .	5
3.3	Complexity . . . . .	5
<b>4</b>	<b>Results . . . . .</b>	<b>6</b>
4.1	Table with Min-Cut results . . . . .	6
4.2	Graph of the Time Cost of the two Algorithms . . . . .	6
4.3	Graph of the Time Cost compared to the Discovery Time of Karger and Stein Algorithm . . . . .	6
4.4	Graph of the Time Cost compared to the Asymptotic Complexity of the two Algorithms . . . . .	6
<b>5</b>	<b>Conclusion . . . . .</b>	<b>7</b>

---

# 1 Introduction

For this assignment, we implemented and analyzed the performance of two algorithms for the min-cut problem for weighted graphs. The algorithms implemented are:

1. Stoer and Wagner's Deterministic Algorithm;
2. Karger and Stein's Randomized Algorithm.

---

## 2 Stoer and Wagner's Deterministic Algorithm

### 2.1 Data Structure

### 2.2 Implementation

### 2.3 Complexity

---

### 3 Karger and Stein's Randomized Algorithm

---

```
1  KARGER (G,k) :
2  A =  $+\infty$ 
3  for i = 1 to k:
4      t = RECURSIVE_CONTRACT(G)
5      if t < min:
6          min = t
7  return min
8
9  RECURSIVE_CONTRACT(G=(D,W)) :
10 n= number of vertices in G
11 if n<=6:
12     Gp= CONTRACT(G,2)
13     return weight of the only edge (u,v) in Gp
14 t =  $n/\sqrt{2}+1$ 
15 for i = 1 to 2:
16     Gi = CONTRACT(G,t)
17     wi = RECURSIVE_CONTRACT(Gi)
18 return min(w1,w2)
19
20 CONTRACT(G=(D,W),k) :
21 n= number of vertices in G
22 for i = 1 to n-k:
23     (u,v) = EDGE_SELECT(D,W)
24     CONTRACT_EDGE(u,v)
25 return D,W
26
27 CONTRACT_EDGE(u,v) :
28 D[u] = D[u]+D[v]-2W[u,v]
29 D[v] = 0
30 W[u,v] = W[v,u] = 0
31 for each vertex w  $\in$  V: except u and v:
32     W[u,v] = W[u,w] + W[v,w]
33     W[w,u] = W[w,u] + W[w,v]
34     W[v,w] = W[w,v] = 0
35
36 EDGE_SELECT(D,W)
37 1. Choose u with probability proportional to D[u]
38 2. Once u is fixed, choose v with probability proportional to W[u,v]
39 3. return the edge (u,v)
```

---

This is a randomized algorithm for the computation of a graph. In the next subsections we explain how we have implemented the data structure and the functions of the algorithm.

---

### 3.1 Data Structure

For the implementation of the algorithm we used:

- **k**: is a constant ( $\log^2 n$ ) used by Karger to repeat Recursive-Contract k times and to obtain an error with probability less or equal to  $1/n$ ;
- **V**: is the list of nodes;
- **D**: is the list of the sum of the weights of each node;
- **W**: is the list of the graph with 3 parameters(node,node,weight). Each node is connected with the others and if they are not connected the third parameter is set to 0 otherwise is set to the correct weight.

### 3.2 Implementation

For the implementation of the algorithm we used these functions:

- **Karger(G,k)**: ;
- **Recursive-Contract(V, W, D)**: ;
- **Contract(s, V, W, D)**: ;
- **Contract-Edge(u,v, W, D)**: ;
- **Edge-Select(V1, D, W)**: that used:
  1. **Random-Select(C)**: ;
  2. **binarySearch(array, x)**: .

### 3.3 Complexity

---

## 4 Results

### 4.1 Table with Min-Cut results

### 4.2 Graph of the Time Cost of the two Algorithms

### 4.3 Graph of the Time Cost compared to the Discovery Time of Karger and Stein Algorithm

### 4.4 Graph of the Time Cost compared to the Asymptotic Complexity of the two Algorithms

---

## 5 Conclusion