

# Brief Article

The Author

November 28, 2013

## Abstract

THIS IS THE ABSTRACT

## Keywords

graph, triangles count

## 1 Introduction

The IMDB dataset provides data about actors, movies and relations between them. Based on this dataset a number of interesting properties can be extracted from the data. The point, however, is doing this in efficient manner.

## 2 Problem Description

The **goal** is to find the three actors, whose movie count they have played together in is maximized among the whole dataset.

**Input** is a list of A-M pair, for each actor A that has played in a movie M.

**Output** is a list of actors with the desired property, on an empty list if no three actors have played together in the same movie.

The algorithm we are presenting works on two main steps:

- Build the data structure - the efficiency of the algorithm is determined by the data structure it runs on. On the other hand, the data structure is specifically designed to solve this problem. Try to solve another problem, say maximizing the number of movies 4 actors have played together in, would require a different structure and approach altogether.
- Traverse data structure and output result - the algorithm works by discarding part of the information on every iteration, thereby reducing the size of the problem.

More details on the algorithm inner workings in the next section.

ALGORITHM SECTION

## 3 Algorithm

The following section covers how the algorithm works. Let us start by defining some notations.

### 3.1 Notations

Let  $G = (V, E)$  be an weighted, undirected simple graph and let  $n = |V|$  and  $m = |E|$ .

A vertex  $v$  denotes an actor. Any edge  $e$  between vertices  $v_1$  and  $v_2$  denotes a set of movies these two actors have played in together. Weight of the edge,  $W(e)$  denotes the size of that set.

Denote by  $A(v)$  the set of adjacent edges to vertex  $v$ .

$YYY(e)$  is the set of (two) vertices adjacent to an edge  $e$ .

$SET(v_1, v_2 \dots v_n)$  - returns a set of unique elements.

MovieCount denotes the biggest number found so far of common movies between any given three actors.

### 3.2 Pseudocode

```
MovieCount  $\leftarrow$  0
Actor1  $\leftarrow$  null
Actor2  $\leftarrow$  null
Actor3  $\leftarrow$  null
for  $v \in V$  do
     $i \leftarrow 0$ 
end for
```

## 4 Analysis

### 4.1 Running time

### 4.2 Space usage

## 5 Conclusion

### 5.1 This is the conclusion text

### 5.2 Future Work

## 6 References