Programming Challenge 2009
Legal Cheating in Elections

Gordon J. Pace, Claudia Borg, Chris Porter

February 2009

Abstract

This document outlines the programming challenge task that has to be solved in the 2009 ICTSA Programming Challenge. The competition will open on Friday, 27th February 2009 at 20:00 and closes on Sunday, 1st March 2009 at 20:00 ZST[1]. This document also specifies the criteria that will be used to judge the entries. The full rules of the Programming Challenge can be found on the website http://programming.ictsamalta.org/

1 Judging Panel

The judging panel is made up of the following three members:

- Claudia Borg, St Martin's Institute of IT, `<claudia.borg@stmartins.edu>`
- Gordon J. Pace, Department of Computer Science, University of Malta `<gordon.pace@um.edu.mt>`
- Chris Porter, Department of Information Systems, University of Malta `<chris.porter@um.edu.mt>`

Any queries to the panel have to be directed to all three members making use of the designated competition Google group which can be accessed by clicking the 'support' button in the navigation menu of the website or, equivalently, by sending an email on programming@ictsamalta.org which will send the email to the google group. No emails sent directly to the judges will be answered. The role of the panel is to judge all submissions made to this competition and declare the winners. The judging panel's decision is final. The formal rules of the competition can be found at the competition's website. Ensure you read through them before you start.

2 The Challenge

2.1 Problem Definition

A country split into villages, each village a known number of supporters of PX, and a known number of supporters of PY . It can be assumed that each village has at least 6 registered voters, and that all registered voters are supporters of one of the two parties. For elections, the country is split into districts, each made up of a number of villages. There are strict rules as to how districts are defined:

- There must be n districts (where n is a given constant for a particular country);

[1]Zeus Standard Time — the time as set on UNIX server zeus, based at the Departments of Computer Science at the University of Malta. This time can be seen on the Programming Challenge website

- Each district is made up of a number of (whole) villages, at least one;
- Each village must lie in exactly one district;
- Villages in a district must be connected — if two villages are in the same district, then you must be able to travel from one to the other travelling only via villages within the district.
- The more uniform the populations of the districts the better.

The input thus consists of (i) a number of villages and, for each village, the voters' preference; (ii) information about which village is adjacent to which other village; (iii) the number of districts to split the country into: n.

The election is quickly approaching, and it is your task to rearrange districts so as to ensure that the outcome is in favour of party PX. How is that possible? For each district, (usually five) members of parliament are elected as follows:

- If the voting population of the district is p, the number of votes required to elect one person in parliament in that district is calculated to be $((p\ \mathit{div}\ 6)+1)$, where *div* is whole number division, ignoring remainder[2]. This is called the quota for that district.
- If q is the quota for the district then, a party which obtains v votes from the villages in the district, elects $v\ \mathit{div}\ q$ members of parliament[3].

Your task is to select districts so as to maximise the number of members of parliament PX manages to elect across the whole country.

2.2 Input File Syntax

Your program will be given a text file consisting of (i) for each village, the votes the parties obtain; (ii) information as to which village is adjacenct to which village; and (iii) the number of districts to split the country into. Village names are assumed to be made up of just letters and numbers (no spaces or other symbols). They are case sensitive, and no limit on their length may be assumed. An input file will look like the following:

```
--villages
...
--connections
...
--districts
...
```

Note that the file is split into three parts:

1. The first part starts with `--villages`, and gives the voting information in the format `villagename:PXvotes,PYvotes`;
2. The second part starts with `--connections`, and gives the connectivity between villages, consisting of a pair of comma-separated village names on each line, indicating that the two villages are adjacent ie order is not important;
3. The last part starts with `--districts`, and has exactly one line giving the number of districts your program should produce.

You may assume that the input files are syntactically correct, and contain no empty lines or extraneous white space. Also, any villages mentioned in the connections part will have appeared in the first part of the file.

[2]For example, $10\ \mathit{div}\ 3 = 3$, $10\ \mathit{div}\ 4 = 2$ and $10\ \mathit{div}\ 5 = 2$.

[3]This sometimes yields only four elected members of parliament in a district. In practice, a more elaborate electoral system would handle these situations.

2.3 Output File Syntax

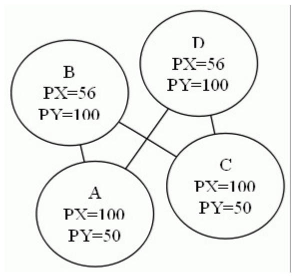
Your program must output a single text file consisting of exactly n lines, where n is the number of districts requested in the input file. Each line will consist of a number of comma separated village names. Apart from the constraints on what constitutes a legal district partitioning given in section 2.1, you must make sure that:

- There are no empty lines;
- No extra white space in the lines;
- No repeated village names within the same line.

2.4 An Example

Consider the following

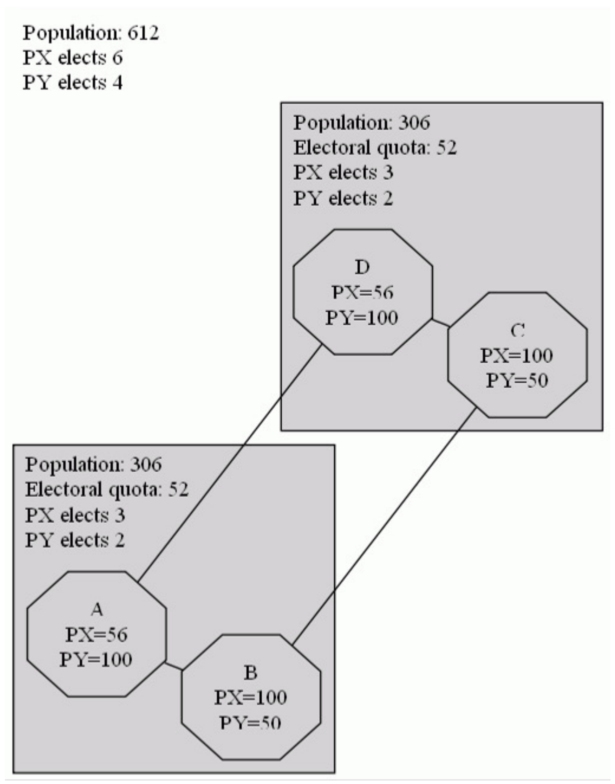
```
--villages
A:56,100
B:100,50
C:100,50
D:56,100
--connections
A,B
B,C
C,D
D,A
--districts
2
```



One solution is to put villages `A` and `B` in one district, and `C` and `D` in another. The output text of the program would thus be:

```
A,B
C,D
```

Visually this would give:

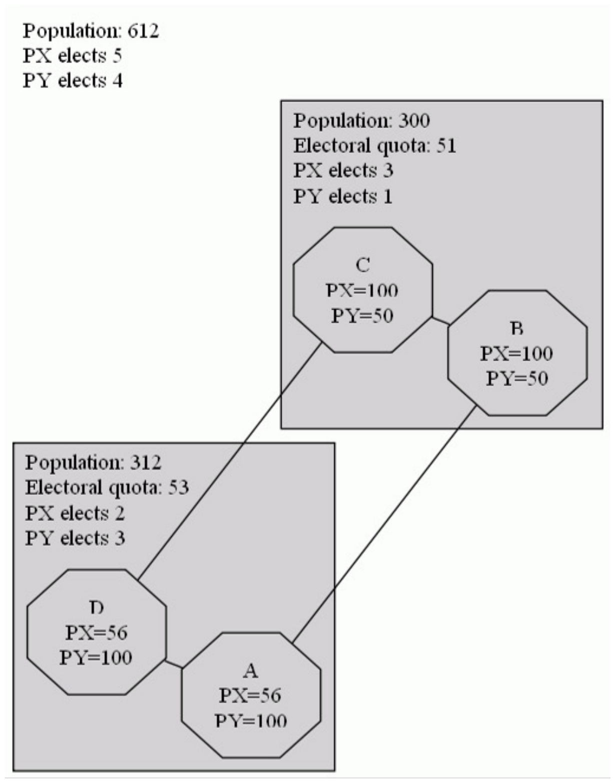


Note that, in total, PX will elect 6 candidates in this district configuration.

Another solution is to put villages `A` and `D` in one district, and `B` and `C` in another. The program's output file will now be:

```
A,D
C,B
```

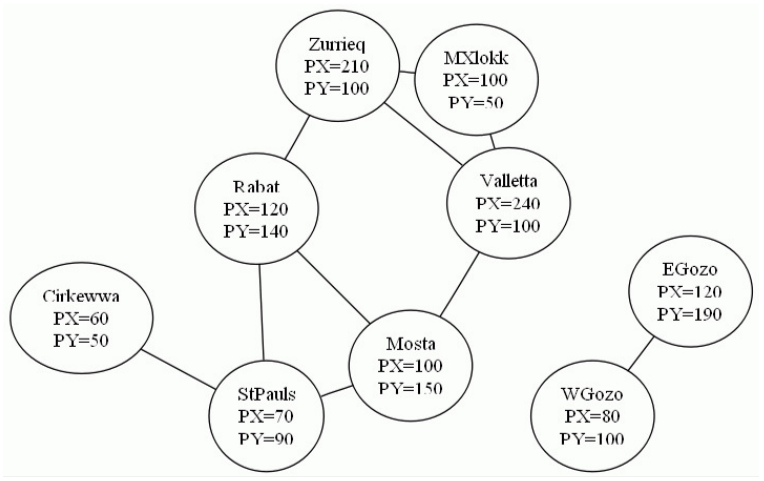
And visually this would give:



In this case, PX will only elect 5 candidates, thus making it inferior.

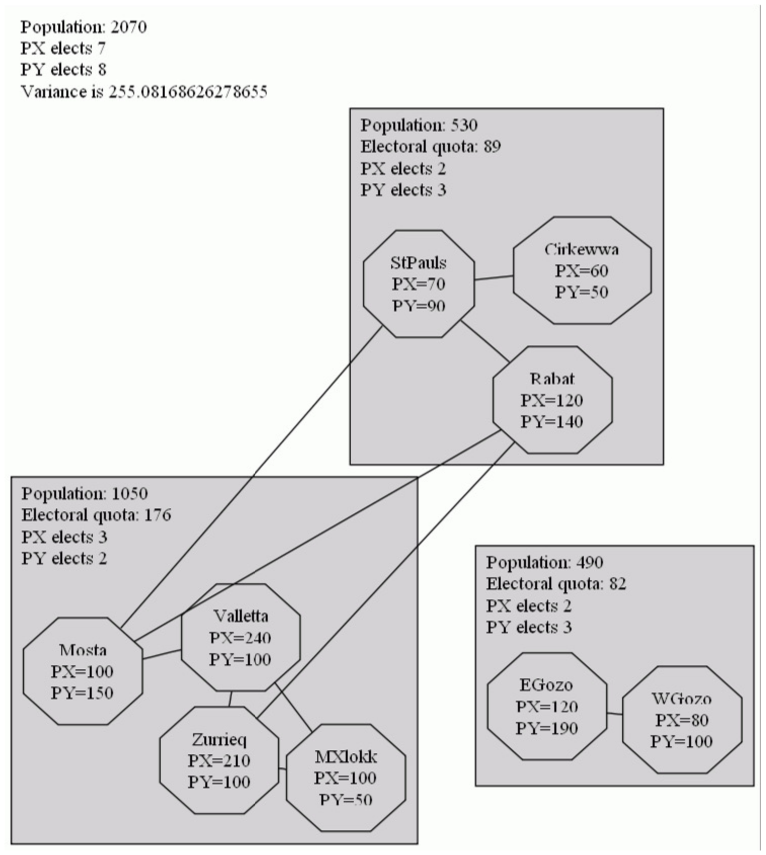
3 A Bigger Example

A bigger example is the one shown in the figure below, where you are asked to produce three districts:



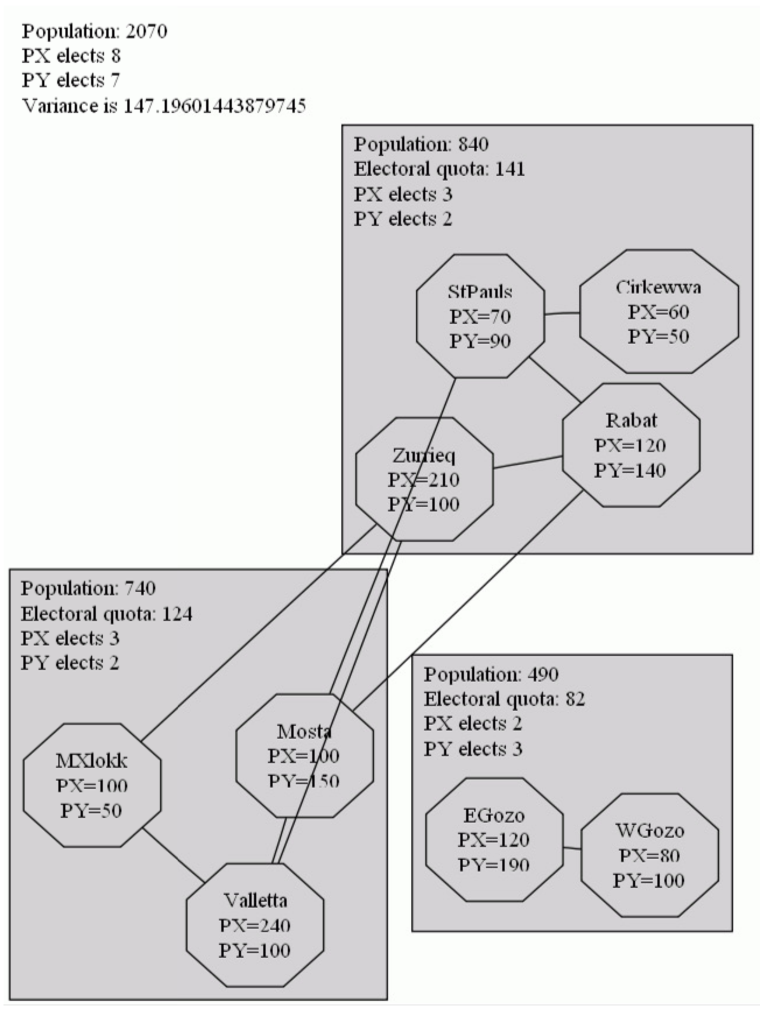
Note that the Gozo villages cannot be joined to any villages in Malta, since they are not connected.

One possible solution to the problem is the following one:

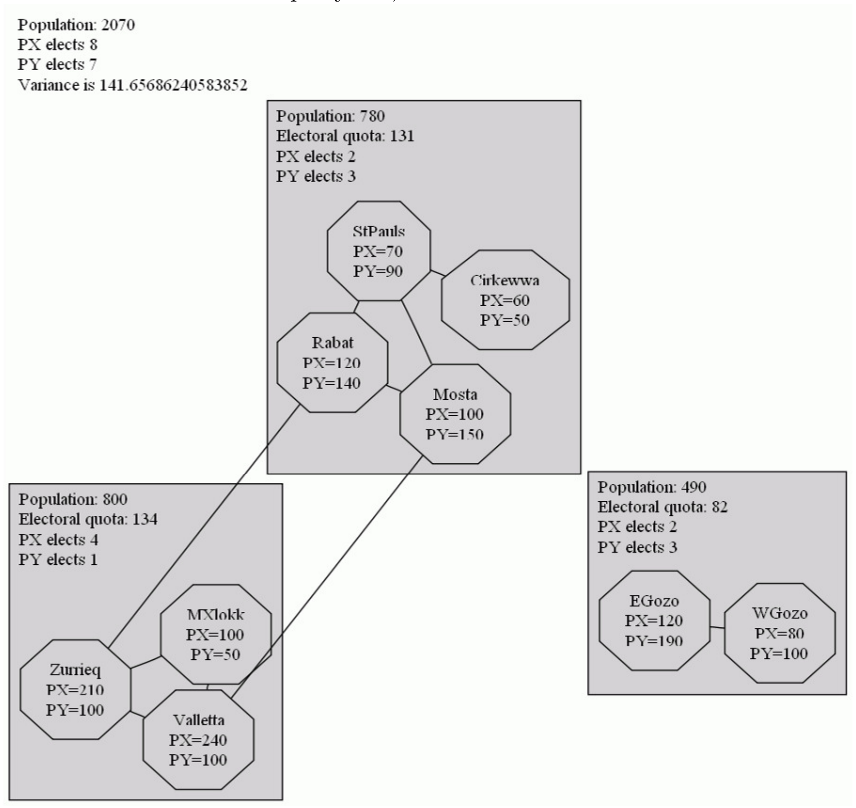


Note that even though Cirkewwa is not direcly connected to Rabat, it is a valid district, since one can go from one to the other without leaving the district.

A better district partitioning is the following:



We can give an even better district layout, which still elects the same number of candidates for party PX, but has a lower variance:



4 Submission Rules

Any submissions have to be a single Microsoft Windows executable file — a command line program called `election.exe` and must take exactly two parameters: the name of the input file and that of the output file. It will be executed as follows:

```
election <input file> <output file>
```

No new windows or graphical user interfaces are to be opened by the program. The program will be executed on an Intel-based PC and having 1GB of RAM. The operating system will be Microsoft Windows XP and it will also have the latest version of the .NET Framework installed. Any submissions that fail to execute on the mentioned environment or any submission that take longer than 5 minutes to output a result will not be considered by the judging team.

Each registered team is allowed to submit as many solutions as it wishes. The last submission on Sunday, 1st March 2009 at 8:00pm will be considered for the prize. After these times no corrections or re-submissions will be allowed. Submissions can only be done through the programming challenge website — note that the program must finish uploading before the aforementioned time.

5 Judging Criteria

The judging panel will run all submissions on a number of pre-defined test problems. For each problem, the programs will be ranked according to the the number of elected members of parliament for party PX (the higher the better) and given a score according to the ranking: best 50 points, second 30 points, third 20 points, fourth 10 points and fifth 5 points. The rest will not be awarded any points. In the case of ties on an individual problem, all submissions solutions with the same number of elected members of parliament for PX will be ranked according to the variance of the populations of the districts[4], the lower, the better. If they still give the same results, they will be given the same ranking for that problem.

The winning team will be the one which obtains the largest total number of points. In the unlikely case of unresolved ties, the ranking will be done according to the execution time of the program over all the problem set (fastest program wins) after which the judging panel reserves the right to resolve any further tie in as fair a way as possible using a tie breaker.

6 Sample Problems and Other Resources

Testing programs: Two programs will be provided to help you test out your programs. The first takes an input file and (optionally) an output solution file, and verifies their correctness, displaying them in textual format. It can be executed on any Windows operating system using the following command line:

```
check <input file> [<solution file>]
```

Any problems with the files' syntax and solution (overlapping districts, missing villages, etc) will be output. If the solution is a valid one, the program also outputs the results of the election.

If only one parameter is given, the program simply outputs the input problem in textual format.

Another program will be provided that is able to generate a dot file[5] from an input file, or an input and solution file for visualization. This can be executed as follows:

```
dotit <input file> [<solution file>] <output dot file>
```

Note that this program also checks whether the solution is valid.

Sample problems: The judging team are providing two sample problems and possible solutions.

The judging suite: To ensure fairness, the problems that will be used for judging the submissions will also be made available when the challenge opens. These will be encrypted and at the end of the competition, the key will be given to ensure that the files were unchanged.

7 Clarifications

During the period between Friday, 27th February at 20:00 and Sunday, 1st March 2009 at 20:00, all the teams can send queries for clarification through the use of the designated competition Google group which can be accessed by clicking the 'support' button in the navigation menu of the website or, equivalently, by sending an email on programming@ictsamalta.org which will send the email to the google group. No emails sent directly to the judges will be answered. If the judging panel deems the query to be valid, the answer will be posted on the group. The judging panel will do its best to answer these queries in as short a time as possible.

[4] If P_i is the population of the ith district, n is the number of districts, and $\overline{P}$ is the average district population, the variance is defined to be $\sqrt{\frac{\sum(P_i - \overline{P})^2}{n}}$.

[5] Read the `README` file provided with the tools to see how dot files can be viewed.