

Flow Report

Anatoli Cooper and Boris Marley

April 20, 2015

Results

Our implementation successfully computes a flow of 163 on the input file, confirming the analysis of the American enemy.¹

We have analysed the possibilities of descreasing the capacities near Minsk. Our analysis is summaries in the following table:²

Case	4W-48	4W-49	Effect on flow
1	30	20	no change
2	20	30	no change
3	20	20	-10
[...]			

In case 4, the new bottleneck becomes

1-44, 3-2, [...]

The comrade from Minsk is advised to [...].

Implementation details

We use a straightforward implemenation of Bellman–Quine’s flow algorithm as described in Bronstein, *Foundations of Algorithms*, chap. 6. We use Prim’s algorithm to find an augmenting path and also implement the neat trick of flow by recomputing the inverse of the antisnail every 2nd iteration.³

The running time is [...].

We have implemented each undirected edge in the input graph as a [...]. In the corresponding residual graph, the edge is represented by a [...]. Our datatype for edge is this: ⁴

```
class FlowEdge
{
    Vertex from, to;
    GUISwingComponentFactory bar;
    Edge next, parent, foo;
}
```

¹ Complete the report by filling in your correct names, filling in the parts marked [...], and changing other parts wherever necessary. Remove the sidenotes in your final hand-in. Insert ‘comrades!’ and ‘for the motherland’ wherever you see fit.

² Complete and correct the table.

³ Replace by a description of what you actually do.

⁴ What I wrote here is complete nonsense, of course. Replace by a description of your actual implementation—what *did* you use? A pair of directed edges? A single undirected edges with a bit to see in which direction the flow goes? As indicated, it makes sense to actually include your datatype for edge in the report. Here, and in general, it makes sense to have written short and sweet code.