Research Institute for Discrete Mathematics
Approximation Algorithms
Summer Term 2013

Prof. Dr. J. Vygen
P. Ochsendorf, M. Sc.

# Programming Exercise

**Exercise P.2:**
Implement the approximation algorithm for 2ECSSP on 2-vertex-connected graphs given in Exercise 9.2.

## Input and Output

**Input** The input graph $G$ (2-edge-connected) will be read from file. The first line contains the numbers $n := |V(G)|$ and $m := |E(G)|$ $(n, m \in \mathbb{N})$ of vertices and edges of $G$. Vertices (edges) are numbered from 0 to $n$ $(m)$. The file contains $m$ further lines describing the edges. Each such line for $e \in E(G)$ contains the two (different) indices of the endpoints of $e \in E(G)$. The $i$th edge will be specified in line $i + 1$. An example might look like this:

```
5 7
1 2
2 3
3 1
0 1
0 3
4 2
3 4
```

Here we have $n = 3$, $m = 7$ and the graph looks as follows:

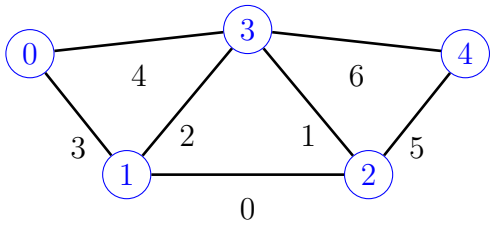


Figure 1: *A sample imput graph: Node indices in blue, edge indices in black.*

**Output** Your program should output the edges of the computed subgraph. The (distinct) indices of $E(H)$ should be printed separated by single spaces in increasing order followed by a newline character. Trailing whitespace should be avoided. In the above example, a correct output would be

```
0 1 3 4 5 6
```

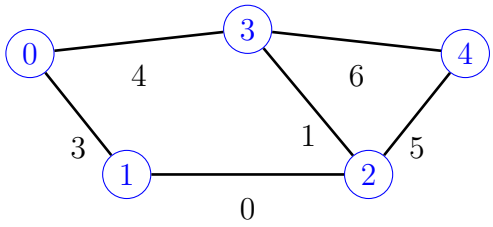


Figure 2: *A correct output for the sample graph.*

**Further Specifications**

- Your code has to compile on linux using `g++/gcc` version 4.4 for `ISO 99 C/C++` with compilerflags `-O2 -pedantic -Wall -Wextra -Wno-long-long` without warnings and errors.
- Third party libraries except from `STL` are prohibited.
- Your code has to run without errors or failures. Additionally it must not result in memory errors as reported by the standard tool `valgrind`.
- In case of nested directories, provide a Makefile for `gnu make` version 3.82.
- You have to document your solution appropriately and safeguard correctly against occuring errors (e.g. during `malloc`).
- Your solution must achieve the proven theoretical running time $O\big(n+m\big)$ in practice.
- Solutions missing out in one or more of the above criteria will not be rewarded with all 16 points even if their output is correct.

**Read in and Instances**

- You may freely use or modify the sample code provided online at `http://www.or.uni-bonn.de/lectures/ss13/approximation_uebung_ss13.html` to read input files meeting the specifications stated above.
- Further instances to test on will be made available online as well.

(16 points)

In case of questions feel free to ask.

Please return your solutions until **June 30th** by mail to `ochsendorf@or.uni-bonn.de`.