Advanced practical **Programming** for Scientists

Thorsten Koch

Zuse Institute Berlin

TU Berlin

14. July 2017

Exercise 6

Using your self written shortest path algorithm, implement the Steiner Tree heuristic by

H. Takahashi, A. Matsuyama

An Approximate Solution for the Steiner Problem in Graphs

Math. Jap. 24 (1980), 573-577

As examples use the graphs from the previous exercise.

Terminals are all nodes which number is prime.

Node Numbers start at 1.

Final Exercise 6b (#8 on github)

Implement the **improved version** of the Steiner Tree heuristic.

Make it run in parallel for given number of starting nodes.

Ex6 filename.gph start_node

Print the edges (as node pairs) of the shortest Steiner tree you find.

Also print the length of the tree.

(Your code should check that what you return is actually a tree and a feasible solution.)

```
/** Computes all primes less than vertexCount */
-> Describe Algorithm shortly
Primes Steiner::getPrimes(unsigned int vertexCount) {
```
-> Why have a general function as part of a specialized class?
```
Primes primes = Primes();
-> Aha!
class Steiner {
   public:
   Steiner();
   virtual ~Steiner();
   void steiner(int vertexCount, Edges* edges,
       Weights& weights, int startnode);
};
-> have verb in procedures name
```

```
class GraphChecker {
  public:
  char isConnected();
  char hasCycle();
}
```

-> char ??? Should be bool.

```
size_t graphSize;
size_t* vertexNo;
size_t*** graph;
```

-> No globals. I doubt *** is needed.

```
char *line = (char*) malloc(MAX_LINE_LEN * sizeof(char));
```

-> cast not needed. Should be sizeof(*line)

Final Exercise

Directory on github: ex10

make THREADS=XX FILE=YYY test should run the test.

Graph files are unpacked at personal directory level SP/*.gph

Output length of the smallest Steiner tree found

TLEN: xxxx

TREE: (from,to) (from,to) ...

TIME: cputime all in seconds and fractions, like 1.637

WALL: wall-clocktime without reading and printing and checking