Advanced Data Structures

Labwork 4

1 Inexact matching

Labwork 1: Consider the strings and `ARCHERY` and `MARCEL`. Indicate a transcript of the first string into the second substring, with minimum number of edit operations.

Note; the transcript is a word made of characters M,D,R, and I.

Labwork 2: Consider the alphabet $\Sigma = \{\mathtt{a}, \mathtt{t}, \mathtt{c}, \mathtt{g}\}$ and the score function

s	a	t	c	g	_
a	1	-1	-2	-4	0
t		0	-3	-2	-1
c			3	0	0
g				2	-2
_					0

Compute the position(s) of the best approximations of $P = \mathtt{atac}$ in the text $T = \mathtt{gatataaac}$.

2 Disjoint set structures

Weighted graphs

A **weighted graph** is a finite set of nodes connected by edges which have positive real numbers as weights. For example, the following is a weighted graph with 5 nodes and 6 edges: We will assume that

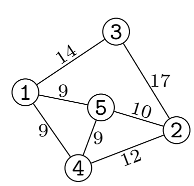


Figure 1: A weighed graph which is connected

- the nodes of a graph with n nodes are labeled with numbers from 1 to n.
- there is a text file which stores the representation of a weighted graph in the following way:
  - The first line contains the value of n (an integer)
  - The following lines contain 3 numbers separated by whitespace:

i j w

    to indicate that the graph has an edge from node i to node j with weight w.

  We assume that the edges are enumerated in increasing order of weight. For example, the weighted graph from Fig. 1 can be stored and read from a text file with the following content:

```
5
1 3 14
1 4 9
1 5 9
4 5 9
2 5 10
2 4 12
2 3 17
```

Kruskal algorithm

Minimum weight spanning trees

A graph is **connected** if there is a path between every two nodes in the graph. Fo example the weighted graph from Fig. 1 is connected.

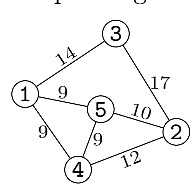
A **spanning tree** of a weighted and connected graph G is a set T of edges of G such that (1) every node of G is an endpoint of an edge in T, and (2) T has no loops. The **weight** $w(T)$ of T is the sum of weights of edges in T.

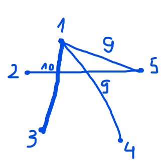
For example, the following are spanning trees of the graph

{1}, {2}, {3}, {4}, {5}
{1,5}, {2}, {3}, {4}
{1,4,5}, {2}, {3}
{1,4,5,2}, {3}
{1,4,5,2,3}- now alg. stops

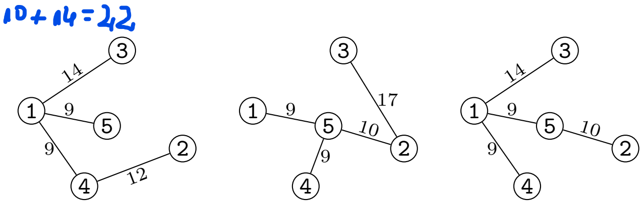
9 1-5 ok
9 1-4 ok
9 4-5 not ok
10 2-5 ok
12 2-4 not ok
14 1-3 ok
17 2-3

not ok = it creates a cycle





is 9+9+10+14=42



$w(T_1) = 44$ $w(T_2) = 45$ $w(T_3) = 42$

cost of it

A **minimum weight spanning tree** (or **MWST**) of G is a spanning tree of G whose weight has minimum possible value. For example, T_3 is a MWST of the graph from Fig. 1.

A MWST of a connected and weighted graph G with n nodes can be found with Kruskal algorithm:

Start with the initial partition $S = \{\{1\}, \{2\}, \dots, \{n\}\}$, $T = \emptyset$ and $W = 0$
for each edge (i, j, w) of G, in increasing order of weights **do**
  if i, j are not in the same component of S
    add (i, j, w) to T
    Union(i, j)
    $W = W + w$
  end if
end for
return T, W

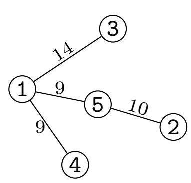
Labwork 3

> This labwork is about using a data structure for disjoint sets to compute a minimum-weight spanning tree of a weighted graph.

Use a data structure for disjoint sets to write a program that reads from a text file `graph.txt` the representation of a connected weighted graph G and computes a MWST of G. The program will print the weight and the list of edges of the MWST.

For example, the output of the program for the graph depicted in Figure 1 can be

```
42
1 4 9
1 5 9
1 3 14
2 5 10
```

to indicate that 

is a MWST.