Tutorial 14

Exercise 1

-

	k=0	k=1	k=2	k=3	k=4	k=5	k=6
a	0	0	0	0	0	0	0
b	∞	7	7	7	7	7	6
c	∞	∞	10	10	10	10	10
d	∞	4	4	4	4	1	1
e	∞	∞	∞	7	7	7	7
f	∞	∞	8	8	-1	-1	-1

- The digraph contains a negative cycle because the columns where $k = 5$ and $k = 6$ are different. It can be easily seen that b, c, e, f, d, b is a cycle with the total cost -1.

Exercise 2

Let us define $max(v) \stackrel{\text{def}}{=} \max_{w \in V}\{d(v,w)\}$ for every vertex $v \in V$.

- **Pre:** The input graph g is a digraph with nonnegative costs on edges and for every edge between two vertices there is an edge in the opposite direction with the same cost.

  $res := hospital(g : DIGRAPH)$

  Post: res is a vertex of the digraph s.t. for every vertex $v \in V$ it holds that $max(res) \leq max(v)$.

- **hospital** (g:DIGRAPH):VERTEX_TYPE =
  $max := \infty$
  $candidate := g.new\_nil\_vertex$
  $v := g.first\_vertex$
  while not $g.nil\_vertex(v)$ **do**
    $dijkstra(g, v)$
    $lmax := 0$
    $u := g.first\_vertex$
    while not $g.nil\_vertex(u)$ **do**
      $lmax :=$ maximum of $u.distance$ and $lmax$
      $u := g.next\_vertex(u)$
    end while
    if $lmax < max$ **then**
      $candidate := v$
      $max := lmax$
    end if
    $v := g.next\_vertex(v)$
  end while
  return $candidate$

- – We have selected adjacency list implementation of the input digraph and hence the operations $g.first\_vertex$, $g.nil\_vertex(v)$ and $g.next\_vertex(u)$ can be performed in $O(1)$ time.
  – As for Dijkstra's algorithm we used Fibonacci heaps based implementation with the worst-case time complexity $O(n \log n + m)$.

  The innermost **while**-loop takes $O(n)$, one call of Dijkstra's algorithm takes $O(n \log n+m)$, the rest in the body of the outermost **while**-loop takes only $O(1)$. The outermost **while**-loop is executed n times hence the total time complexity is $n \cdot \big(O(n \log n+m)+O(n)+O(1)\big) = n \cdot O(n \log n+m) = O(n^2 \log n+nm)$.

Exercise 3

Prof. *Konfus* is again wrong. Dijkstra's algorithm called from the start node a will give a wrong answer e.g. on the following digraph:

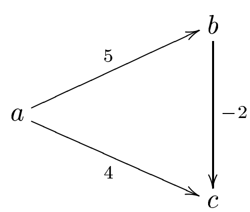
- $G = (V, E)$, where
- $V = \{a, b, c\}$ and

  $E = \{(a, b), (a, c), (b, c)\}$ with the following costs on edges:

  · $c(a, b) = 5$,

  · $c(a, c) = 4$ and

  · $c(b, c) = -2$.



The shortest distance between a and c is $d(a, c) = 3$; Dijkstra's algorithm returns 4.