

# ODABRANA POGLAVLJA PROJEKTOVANJA FIZIČKE ARHITEKTURE

## PROJEKTOVANJE ALGORITAMA – TEST 4

1 Jun 2019

### ZADATAK

Na osnovu priloženih tabela rutiranja potrebno formirati topologiju mreže i pomoću njega pronaći najjeftiniju putanju slanja paketa u konkretnoj mreži. U nastavku su date tabele rutiranja čvorova u mreži.

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.242.100</b>	5
<b>192.168.242.73</b>	3

Tabela rutiranja čvora sa IP adresom  
192.168.242.10

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.242.95</b>	4
<b>192.168.242.102</b>	3

Tabela rutiranja čvora sa IP adresom  
192.168.242.100

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.242.84</b>	3

Tabela rutiranja čvora sa IP adresom  
192.168.242.95

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.10.11</b>	1

Tabela rutiranja čvora sa IP adresom  
192.168.10.10

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.242.102</b>	3
<b>192.168.242.48</b>	1

Tabela rutiranja čvora sa IP adresom  
192.168.242.35

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.242.84</b>	4
<b>192.168.242.73</b>	1

Tabela rutiranja čvora sa IP adresom  
192.168.242.48

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.242.84</b>	1
<b>192.168.242.10</b>	1

Tabela rutiranja čvora sa IP adresom  
192.168.242.102

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
<b>192.168.10.10</b>	1

Tabela rutiranja čvora sa IP adresom  
192.168.10.11

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
192.168.242.35	4

Tabela rutiranja čvora sa IP adresom  
192.168.242.73

Odredišni čvor	Trošak slanja paketa ka odredišnom čvoru
192.168.242.100	1

Tabela rutiranja čvora sa IP adresom  
192.168.242.84

1. Napisati funkciju *MakeGraph* koja formira graf koji modeluju gore priložene tabele rutiranja.  
`def MakeGraph() --> returns graph`
2. Napisati funkciju *ShortestPath* koja računa najkraću putanju od čvora **src** do čvora **dst** koristeći optimalan algoritam pretrage.  
`def ShortestPath(src, dst, graph) --> returns (path, int)`

Povratni parametar *path* je tipa string u sledećem formatu: *src->node1->node2-> ... -> dst*

3. Napisati funkciju *UpdateEdge* koja dodaje ivicu između dva čvora sa težinom koja se prosleđuje kao ulazni parametar. Ukoliko ivica već postoji između ta dva čvora, ova funkcija menja težinu date ivice.  
`def UpdateEdge(graph, nodeA, nodeB, weight) --> returns graph`