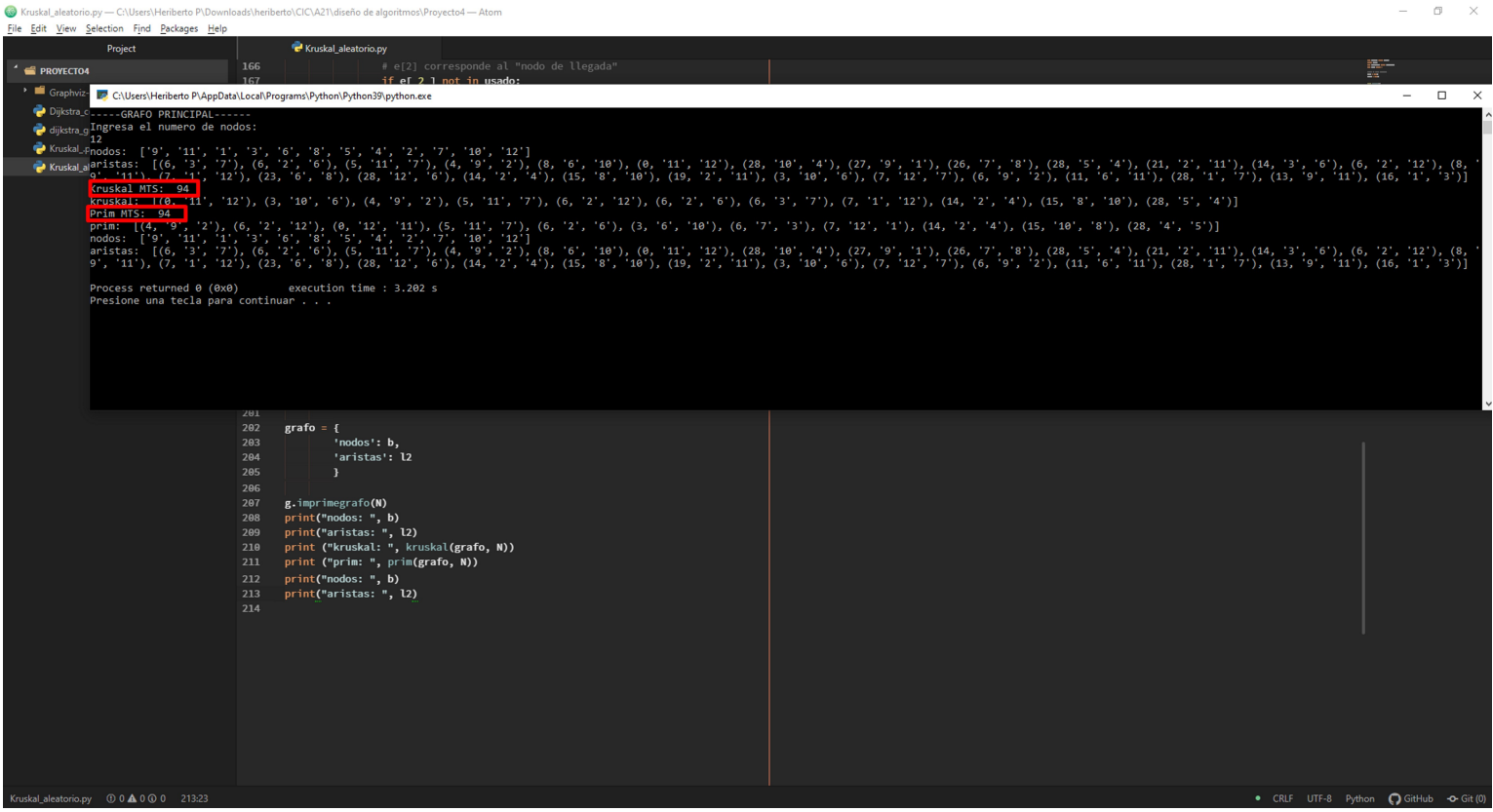
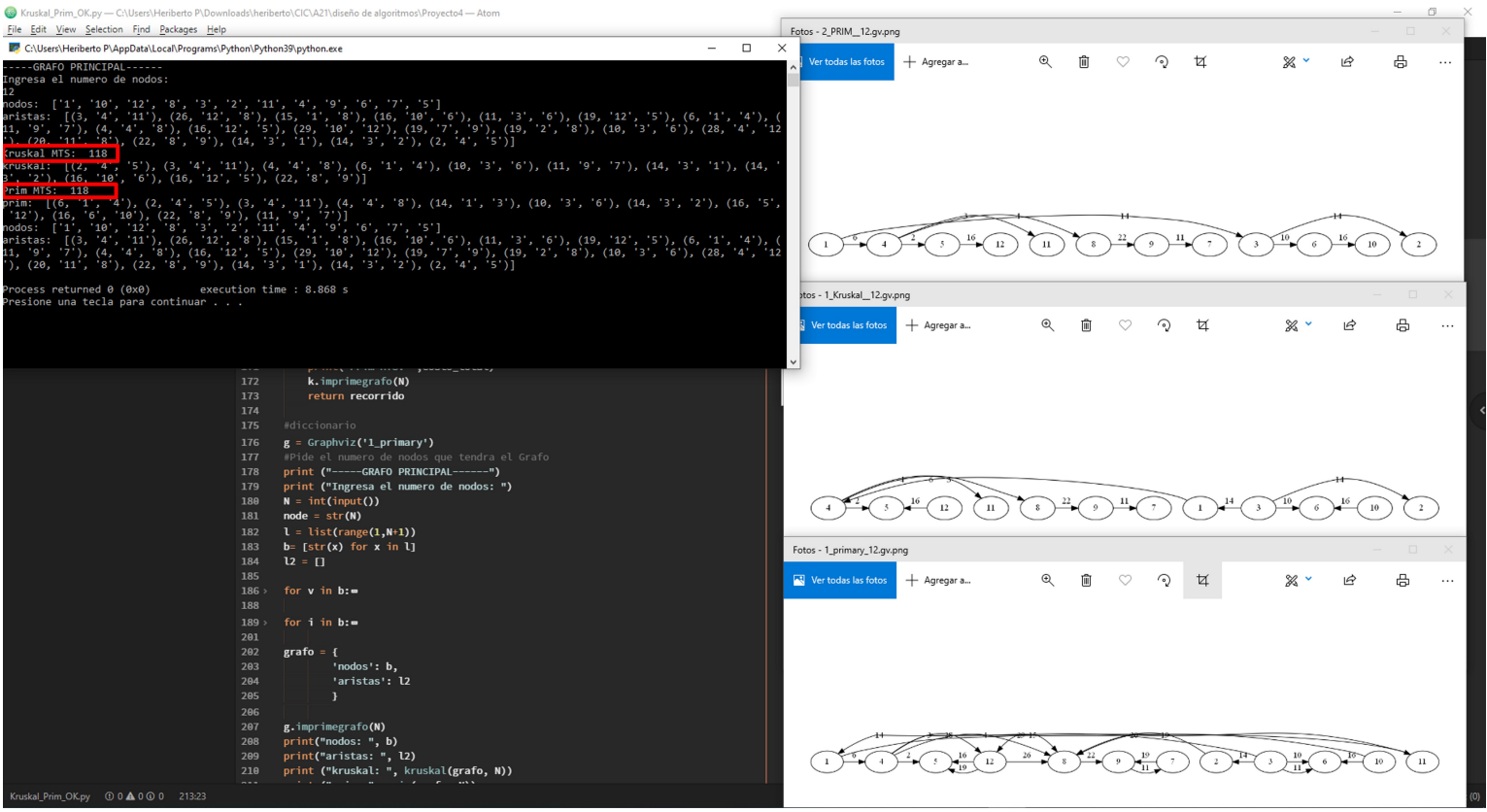
Proyecto 4 Capturas de Pantalla

miércoles, 23 de junio de 2021 05:01 p. m.





Kruskal_Prim_OK.py — C:\Users\Heriberto P\Downloads\heriberto\CIC\A21\diseño de algoritmos\Proyecto4 — Atom

File Edit View Selection Find Packages Help

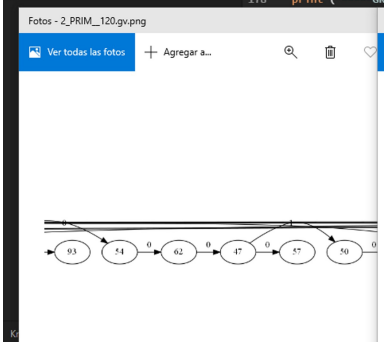
Project | Kruskal_Prim_OK.py

C:\Users\Heriberto P\AppData\Local\Programs\Python\Python39\python.exe

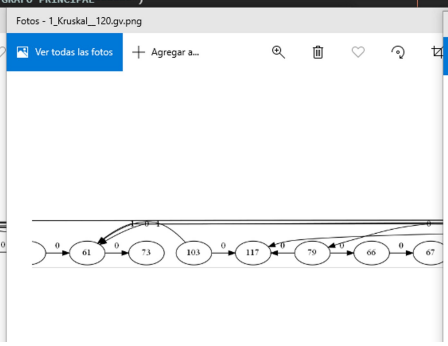
```
5'), (3, '98', '113'), (16, '65', '15'), (27, '5', '82'), (17, '89', '14'), (18, '105', '16'), (17, '58', '78'), (28, '31', '61'), (23, '116', '101'), (1, '18', '118'), (6, '114', '90'), (14, '7', '46'), (20, '10', '87'), (9, '94', '43'
), (7, '112', '24'), (2, '85', '50'), (26, '67', '29'), (21, '51', '93'), (21, '38', '47'), (10, '82', '53'), (12, '88', '99'), (15, '72', '105'), (8, '100', '119'), (29, '49', '17'), (15, '63', '78'), (20, '66', '69'), (17, '8', '11'),
 (24, '91', '71'), (19, '40', '105'), (22, '101', '113'), (7, '69', '18'), (6, '14', '94'), (25, '60', '106'), (14, '41', '74'), (12, '9', '19'), (0, '47', '62'), (11, '66', '6'), (25, '99', '120'), (18, '82', '108'), (29, '63', '49'),
 (17, '21', '80'), (2, '28', '43'), (10, '25', '114'), (26, '51', '93'), (21, '31', '90'), (19, '76', '2')]
Kruskal MTS:  77
kruskal:  [(0, '1', '111'), (0, '1', '29'), (0, '100', '102'), (0, '101', '15'), (0, '103', '117'), (0, '115', '1'), (0, '115', '43'), (0, '115', '54'), (0, '118', '46'), (0, '119', '29'), (0, '14', '70'), (0, '14', '86'), (0, '16', '29
'), (0, '17', '29'), (0, '17', '63'), (0, '19', '109'), (0, '20', '38'), (0, '20', '95'), (0, '22', '8'), (0, '27', '34'), (0, '27', '51'), (0, '3', '5'), (0, '30', '79'), (0, '32', '45'), (0, '36', '85'), (0, '37', '59'), (0, '37', '68
'), (0, '38', '3'), (0, '47', '62'), (0, '49', '30'), (0, '49', '38'), (0, '50', '53'), (0, '52', '61'), (0, '54', '62'), (0, '55', '60'), (0, '57', '47'), (0, '58', '106'), (0, '6', '110'), (0, '60', '113'), (0, '61', '73'), (0, '63',
'61'), (0, '65', '14'), (0, '65', '89'), (0, '66', '67'), (0, '67', '15'), (0, '67', '94'), (0, '68', '92'), (0, '69', '111'), (0, '7', '20'), (0, '7', '48'), (0, '72', '38'), (0, '79', '117'), (0, '79', '66'), (0, '84', '118'), (0, '86
', '77'), (0, '87', '27'), (0, '88', '44'), (0, '89', '82'), (0, '91', '39'), (0, '94', '18'), (1, '1', '84'), (1, '100', '82'), (1, '103', '61'), (1, '108', '84'), (1, '11', '75'), (1, '113', '4'), (1, '115', '87'), (1, '116', '16'), (
1, '12', '113'), (1, '12', '45'), (1, '120', '26'), (1, '14', '24'), (1, '15', '58'), (1, '16', '107'), (1, '16', '89'), (1, '18', '12'), (1, '19', '18'), (1, '23', '27'), (1, '26', '22'), (1, '27', '39'), (1, '29', '78'), (1, '30', '21
'), (1, '31', '105'), (1, '31', '3'), (1, '34', '44'), (1, '36', '52'), (1, '37', '119'), (1, '40', '10'), (1, '40', '117'), (1, '41', '58'), (1, '47', '50'), (1, '5', '98'), (1, '51', '35'), (1, '52', '22'), (1, '64', '92'), (1, '69',
'81'), (1, '7', '13'), (1, '75', '109'), (1, '75', '74'), (1, '76', '4'), (1, '80', '10'), (1, '83', '119'), (1, '93', '43'), (1, '95', '71'), (1, '96', '61'), (2, '104', '32'), (2, '107', '42'), (2, '21', '97'), (2, '25', '85'), (2, '2
8', '35'), (2, '33', '111'), (2, '56', '116'), (2, '80', '110'), (2, '9', '56'), (2, '90', '89'), (3, '114', '68'), (3, '2', '119'), (3, '26', '112'), (3, '78', '99')]
Prim MTS:  77
prim:  [(1, '40', '10'), (1, '10', '40'), (1, '10', '80'), (1, '40', '117'), (0, '117', '103'), (0, '117', '79'), (0, '79', '30'), (0, '30', '49'), (0, '49', '38'), (0, '38', '20'), (0, '20', '7'), (0, '20', '94'), (0, '20', '95'), (0,
'38', '3'), (0, '3', '5'), (0, '38', '72'), (0, '7', '48'), (0, '79', '66'), (0, '66', '67'), (0, '67', '15'), (0, '15', '101'), (0, '94', '18'), (1, '103', '61'), (0, '61', '52'), (0, '61', '63'), (0, '61', '73'), (0, '63', '111'), (0,
'111', '1'), (0, '1', '115'), (0, '1', '29'), (0, '111', '69'), (0, '115', '42'), (0, '115', '54'), (0, '29', '119'), (0, '29', '16'), (0, '29', '17'), (0, '54', '62'), (0, '62', '47'), (0, '47', '57'), (1, '1', '84'), (0, '84', '118'),
 (0, '118', '46'), (1, '115', '87'), (0, '87', '27'), (0, '27', '34'), (0, '27', '51'), (1, '119', '37'), (0, '37', '59'), (0, '37', '68'), (0, '68', '92'), (1, '119', '83'), (1, '15', '58'), (0, '58', '106'), (1, '16', '107'), (1, '16
', '116'), (1, '116', '60'), (0, '60', '113'), (0, '60', '55'), (1, '113', '12'), (1, '12', '45'), (0, '45', '32'), (1, '16', '89'), (0, '89', '65'), (0, '65', '14'), (0, '14', '70'), (0, '14', '86'), (0, '86', '77'), (0, '89', '82'), (
1, '14', '24'), (1, '18', '19'), (0, '19', '109'), (1, '109', '75'), (1, '27', '23'), (1, '27', '39'), (0, '39', '91'), (1, '29', '78'), (1, '3', '31'), (1, '30', '21'), (1, '31', '105'), (1, '34', '44'), (0, '44', '88'), (1, '43', '93
'), (1, '47', '50'), (0, '50', '53'), (1, '5', '98'), (1, '51', '35'), (1, '52', '22'), (0, '22', '8'), (1, '22', '26'), (1, '26', '120'), (1, '52', '36'), (0, '36', '85'), (1, '58', '41'), (1, '61', '96'), (1, '69', '81'), (1, '7', '13
'), (1, '75', '11'), (1, '75', '74'), (1, '75', '76'), (1, '82', '100'), (0, '100', '102'), (1, '84', '108'), (1, '92', '64'), (1, '95', '71'), (2, '107', '42'), (2, '111', '33'), (2, '116', '56'), (2, '21', '97'), (2, '32', '104'), (2,
'35', '28'), (2, '56', '9'), (2, '80', '110'), (0, '110', '6'), (2, '85', '25'), (2, '89', '90'), (3, '119', '2'), (3, '26', '112'), (3, '68', '114'), (3, '78', '99')]
nodos:  ['40', '105', '101', '113', '69', '18', '14', '94', '60', '106', '41', '74', '9', '19', '47', '62', '66', '6', '99', '120', '82', '108', '63', '49', '72', '80', '28', '43', '25', '114', '51', '93', '31', '90', '76', '2', '64',
'117', '39', '61', '85', '23', '118', '30', '54', '8', '46', '50', '75', '44', '98', '68', '36', '56', '112', '37', '104', '35', '20', '16', '22', '1', '24', '3', '21', '65', '32', '83', '110', '71', '33', '119', '89', '12', '78', '100',
'11', '15', '115', '107', '45', '42', '87', '111', '5', '4', '116', '81', '34', '57', '96', '86', '59', '29', '67', '91', '58', '38', '13', '88', '48', '97', '102', '95', '55', '7', '84', '27', '52', '109', '17', '53', '103', '92', '26
', '70', '79', '73', '77', '10']
aristas:  [(13, '30', '10'), (24, '111', '116'), (13, '57', '85'), (26, '4', '23'), (17, '120', '37'), (11, '32', '68'), (6, '63', '58'), (9, '35', '70'), (0, '19', '109'), (14, '23', '60'), (18, '79', '45'), (28, '21', '51'), (8, '30',
'54'), (24, '90', '67'), (8, '58', '74'), (20, '28', '33'), (22, '98', '74'), (10, '51', '70'), (8, '58', '49'), (24, '4', '99'), (26, '40', '24'), (0, '57', '47'), (11, '34', '92'), (17, '6', '117'), (27, '79', '64'), (19, '44', '48')]
```

```
178 print ("------GRAFO PRINCIPAL-------")
```

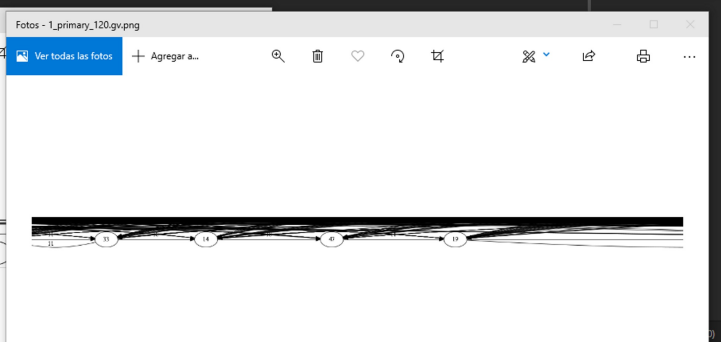
Fotos - 2_PRIM_120.gv.png



Fotos - 1_Kruskal_120.gv.png



Fotos - 1_primary_120.gv.png



Kruskal_Prim_OK.py — C:\Users\Heriberto P\Downloads\heriberto\CIC\A21\diseño de algoritmos\Proyecto4 — Atom

File Edit View Selection Find Packages Help

Project

PROYECTO4

- Graphviz-output
- Dijkstra_complete.py
- dijkstra_graphviz.py
- Kruskal_.py
- Kruskal_aleatorio.py
- Kruskal_Prim_OK.py

Kruskal_Prim_OK.py

```
130 #BUSCAR ARBOL MINIMO
131
132 def prim( grafo, N ):
133     k = Graphviz("2_PRIM_")
134     #crea listas en base a un indice comun, en este caso los indices seran los nodos 1 y 2
135     #en cada indice se almacena la tupla (c, n1, n2)
136     conn = defaultdict( list )
137     for c,n1,n2 in grafo['aristas']:
138         #direccionamiento
139         conn[ n1 ].append( (c, n1, n2) )
140         conn[ n2 ].append( (c, n2, n1) )
141
142     recorrido = []
143     lcostos = []
144     #toma el nodo inicial
145     usado = set( grafo['nodos'][0] )
146     #toma las aristas que contienen el nodo inicial
147     nueva_arista = conn[ grafo['nodos'][0] ][:]
148
149     #mantiene en la posicion 0 el menor valor de la lista
150     heapify( nueva_arista )
151
152     while nueva_arista:
153         #saca el primer valor de la lista y lo almacena en costo, n1, n2
154         costo, n1, n2 = heappop( nueva_arista )
155         #pregunta si el nodo final de la arista no ha sido visitado
156         if n2 not in usado:
157             usado.add( n2 )
158             #agrega la arista al recorrido
159             recorrido.append( ( costo, n1, n2  ) )
160             k.agregaedge(n1, n2, costo)
161             lcostos.append(costo)
162             #print "recorrido",recorrido
163
164             #recorre la lista de nodos invertidos y en caso de que no se aya pasado por el nodo lo agrega a la lista de aristas.
165             for e in conn[ n2 ]:
166                 # e[2] corresponde al "nodo de llegada"
167                 if e[ 2 ] not in usado:
168                     #agrega "e" a nueva_arista
169                     heappush( nueva_arista, e )
170     costo_total = sum(lcostos)
171     print("Prim MTS: ",costo_total)
172     k.imprimegrafo(N)
173     return recorrido
174
175 #diccionario
176 g = Graphviz('1_primary')
```

Kruskal_Prim_OK.py 0 0 0 0 213:23 CRLF UTF-8 Python GitHub Git (0)