

INF-354-2P-P3

November 26, 2023

1 Segundo Examen Parcial INF - 354

1.0.1 Nombre: Steve Brandom Nina Huacani

1.1 Pregunta N°3: Dado el siguiente grafo:

Generar el archivo CSV, obtener el mejor recorrido usando algoritmos genéticos sin el uso de DEAP.

```
[1]: #Importamos la libreria numpy
import numpy as np
#importamos la libreria para generar numeros aleatorios
import random

[2]: #Definimos el numero de nodos del grafo
n_nodos = 5
#Definimos el tamaño de la poblacion
poblacion = 50
#Probabilidad de mutacion
probabilidad_mutacion = 0.2
#Definimos una lista de nodos
lista_nodos = np.array(['A', 'B', 'C', 'D', 'E'])
#Creamos un diccionario con las distancias entre cada nodo
diccionario_distancias = {'AB':7, 'BA':7, 'AC':9, 'CA':9, 'CE':5, 'EC':5, 'ED':
    ↪17, 'DE':17, 'BD':4, 'DB':4, 'DA':8, 'AD':8, 'AE':20, 'EA':20, 'BC':10, 'CB':
    ↪10, 'CD':15, 'DC':15, 'BE':11, 'EB':11}

[3]: #creamos una funcion para calcular la distancia entre dos nodos
def distancia_entre_nodos(x, y, diccionario_distancias):
    #retornamos el valor de la distancia entre esos puntos
    return diccionario_distancias.get(x+y)

#Definimos la funcion para tener las combinaciones de las posibles soluciones
def posibles_soluciones(lista_nodos, poblacion):
    #Creamos un array para almacenar las posibles soluciones
    soluciones = []
    #iteramos
    for i in range(poblacion):
        #elegimos aleatoriamente permutaciones entre los nodos de la lista
```

```

    solucion = lista_nodos[np.random.choice(list(range(n_nodos)), n_nodos,
↪replace=False)]
    #añadimos la solucion a un array que sera nuestras posibles soluciones
    soluciones.append(solucion)
    #retornamos el array de las posibles soluciones
    return np.array(soluciones)

#obtenemos el conjunto inicial de soluciones
conjunto_soluciones = posibles_soluciones(lista_nodos, poblacion)
print(conjunto_soluciones)

```

```

[['E' 'B' 'D' 'A' 'C']
 ['D' 'E' 'B' 'C' 'A']
 ['C' 'D' 'B' 'E' 'A']
 ['E' 'A' 'D' 'B' 'C']
 ['C' 'D' 'E' 'B' 'A']
 ['B' 'A' 'D' 'C' 'E']
 ['E' 'D' 'A' 'B' 'C']
 ['C' 'B' 'D' 'E' 'A']
 ['E' 'A' 'D' 'B' 'C']
 ['C' 'D' 'A' 'E' 'B']
 ['C' 'A' 'B' 'D' 'E']
 ['E' 'C' 'B' 'D' 'A']
 ['D' 'A' 'B' 'C' 'E']
 ['A' 'B' 'D' 'E' 'C']
 ['C' 'A' 'B' 'D' 'E']
 ['B' 'C' 'E' 'A' 'D']
 ['E' 'A' 'B' 'C' 'D']
 ['A' 'D' 'B' 'E' 'C']
 ['A' 'C' 'D' 'E' 'B']
 ['C' 'E' 'D' 'B' 'A']
 ['C' 'D' 'E' 'B' 'A']
 ['D' 'B' 'A' 'E' 'C']
 ['E' 'B' 'A' 'D' 'C']
 ['E' 'C' 'B' 'A' 'D']
 ['C' 'D' 'E' 'A' 'B']
 ['C' 'B' 'D' 'A' 'E']
 ['E' 'A' 'B' 'C' 'D']
 ['D' 'A' 'B' 'E' 'C']
 ['D' 'E' 'B' 'C' 'A']
 ['E' 'B' 'A' 'C' 'D']
 ['C' 'E' 'A' 'B' 'D']
 ['D' 'B' 'C' 'E' 'A']
 ['E' 'D' 'B' 'A' 'C']
 ['A' 'D' 'E' 'B' 'C']
 ['B' 'E' 'C' 'D' 'A']
 ['A' 'B' 'C' 'E' 'D']
 ['D' 'A' 'E' 'C' 'B']

```

```

['A' 'E' 'B' 'D' 'C']
['E' 'B' 'D' 'C' 'A']
['D' 'E' 'B' 'C' 'A']
['B' 'A' 'E' 'D' 'C']
['A' 'C' 'E' 'B' 'D']
['E' 'C' 'D' 'B' 'A']
['C' 'D' 'E' 'A' 'B']
['C' 'E' 'B' 'D' 'A']
['E' 'B' 'D' 'A' 'C']
['C' 'A' 'D' 'B' 'E']
['E' 'A' 'D' 'C' 'B']
['D' 'A' 'E' 'C' 'B']
['B' 'C' 'E' 'A' 'D']

```

```

[4]: #Definimos la funcion para evaluar la aptitud de una posible solucion
def evaluar_aptitud(lista_nodos, diccionario_distancias):
    #iniciamos la variable que almacenara el costo del viaje
    total = 0
    #iteramos sobre los nodos
    for i in range(n_nodos-1):
        #Extraemos un nodo
        nodo_a = lista_nodos[i]
        #extraemos otro nodo
        nodo_b = lista_nodos[i+1]
        #sumamos el costo de nodo a nodo
        total += distancia_entre_nodos(nodo_a, nodo_b, diccionario_distancias)
    #Sumamos el valor de vuelta al nodo
    total += distancia_entre_nodos(lista_nodos[0], lista_nodos[n_nodos-1],
    ↪diccionario_distancias)
    #retornamos el total del costo del viaje
    return total

#definimos la funcion para evaluar la aptitud de cada posible solucion
def evaluar_posibles_soluciones(conjunto_soluciones, diccionario_distancias):
    #definimos una lista
    lista = np.zeros(poblacion)
    #iteramos sobre la cantidad de posibles soluciones
    for i in range(poblacion):
        #almacenamos el costo total del viaje
        lista[i] = evaluar_aptitud(conjunto_soluciones[i], diccionario_distancias)
    #retornamos la lista de costos
    return lista

#obtenemos la lista de costos
lista_costos = evaluar_posibles_soluciones(conjunto_soluciones,
    ↪diccionario_distancias)
print(lista_costos)

```

```
[37. 55. 59. 47. 59. 46. 47. 60. 47. 64. 42. 47. 47. 42. 42. 47. 69. 37.
 59. 42. 59. 51. 46. 47. 69. 47. 69. 46. 55. 59. 51. 47. 42. 55. 46. 47.
 47. 59. 59. 55. 69. 37. 51. 69. 37. 37. 37. 64. 47. 47.]
```

```
[5]: #definimos la funcion para elegir al progenitor
def seleccion_progenitor(conjunto_soluciones, lista_costos):
    #obtenemos el costo total de las soluciones
    total_costos = lista_costos.sum()
    #calcula la probabilidad de seleccion de progenitor Esto normaliza las
    #aptitudes para que sumen 1 y crea una distribución de probabilidad.
    probabilidades = lista_costos/total_costos
    #Utiliza la distribución de probabilidad para seleccionar índices de
    ↪soluciones de la población
    #omo progenitores. La selección se realiza aleatoriamente
    #pero las soluciones con mayores aptitudes tienen una mayor probabilidad de
    ↪ser seleccionadas.
    lista_progenitores_a = np.random.
    ↪choice(list(range(len(conjunto_soluciones))), len(conjunto_soluciones),
    ↪p=probabilidades, replace=True)
    #Realiza la misma selección de progenitores para la segunda lista de
    ↪progenitores.
    lista_progenitores_b = np.random.
    ↪choice(list(range(len(conjunto_soluciones))), len(conjunto_soluciones),
    ↪p=probabilidades, replace=True)
    #Obtenemos las soluciones reales correspondientes a los índices seleccionados
    ↪para los progenitores.
    lista_progenitores_a = conjunto_soluciones[lista_progenitores_a]
    #realizamos lo mismo con el otro conjunto de progenitores
    lista_progenitores_b = conjunto_soluciones[lista_progenitores_b]
    #Devolvemos un array que contiene las dos listas de progenitores.
    return np.array([lista_progenitores_a, lista_progenitores_b])

#obtenemos la lista de progenitores
lista_progenitores = seleccion_progenitor(conjunto_soluciones, lista_costos)
print(lista_progenitores)
```

```
[[['A' 'E' 'B' 'D' 'C']
  ['B' 'A' 'E' 'D' 'C']
  ['E' 'A' 'D' 'B' 'C']
  ['E' 'A' 'D' 'B' 'C']
  ['D' 'B' 'C' 'E' 'A']
  ['E' 'B' 'A' 'D' 'C']
  ['A' 'E' 'B' 'D' 'C']
  ['C' 'B' 'D' 'E' 'A']
  ['C' 'D' 'E' 'A' 'B']
  ['C' 'D' 'B' 'E' 'A']
  ['E' 'A' 'B' 'C' 'D']
  ['E' 'C' 'B' 'D' 'A']]
```

['C' 'E' 'A' 'B' 'D']
 ['D' 'E' 'B' 'C' 'A']
 ['D' 'E' 'B' 'C' 'A']
 ['C' 'B' 'D' 'E' 'A']
 ['E' 'C' 'B' 'A' 'D']
 ['E' 'C' 'D' 'B' 'A']
 ['E' 'B' 'D' 'A' 'C']
 ['E' 'A' 'D' 'B' 'C']
 ['E' 'A' 'D' 'B' 'C']
 ['D' 'B' 'C' 'E' 'A']
 ['C' 'A' 'B' 'D' 'E']
 ['E' 'C' 'B' 'D' 'A']
 ['A' 'B' 'C' 'E' 'D']
 ['E' 'A' 'D' 'B' 'C']
 ['D' 'B' 'C' 'E' 'A']
 ['C' 'B' 'D' 'A' 'E']
 ['E' 'B' 'D' 'C' 'A']
 ['A' 'C' 'D' 'E' 'B']
 ['C' 'A' 'B' 'D' 'E']
 ['B' 'A' 'E' 'D' 'C']
 ['C' 'B' 'D' 'A' 'E']
 ['C' 'A' 'B' 'D' 'E']
 ['A' 'C' 'D' 'E' 'B']
 ['C' 'D' 'E' 'A' 'B']
 ['D' 'B' 'A' 'E' 'C']
 ['D' 'E' 'B' 'C' 'A']
 ['D' 'A' 'E' 'C' 'B']
 ['C' 'D' 'A' 'E' 'B']
 ['E' 'B' 'D' 'C' 'A']
 ['D' 'B' 'A' 'E' 'C']
 ['A' 'E' 'B' 'D' 'C']
 ['E' 'C' 'D' 'B' 'A']
 ['B' 'A' 'E' 'D' 'C']
 ['B' 'E' 'C' 'D' 'A']
 ['E' 'A' 'D' 'C' 'B']
 ['E' 'B' 'A' 'C' 'D']
 ['D' 'B' 'A' 'E' 'C']
 ['C' 'D' 'E' 'A' 'B']]

[['D' 'A' 'B' 'C' 'E']
 ['A' 'C' 'E' 'B' 'D']
 ['E' 'D' 'B' 'A' 'C']
 ['C' 'D' 'E' 'A' 'B']
 ['B' 'A' 'E' 'D' 'C']
 ['C' 'D' 'E' 'A' 'B']
 ['A' 'D' 'E' 'B' 'C']
 ['B' 'C' 'E' 'A' 'D']
 ['C' 'E' 'A' 'B' 'D']

```

['D' 'A' 'B' 'C' 'E']
['E' 'A' 'B' 'C' 'D']
['D' 'A' 'B' 'E' 'C']
['C' 'D' 'E' 'A' 'B']
['C' 'D' 'E' 'A' 'B']
['E' 'A' 'B' 'C' 'D']
['D' 'B' 'C' 'E' 'A']
['D' 'B' 'A' 'E' 'C']
['C' 'A' 'B' 'D' 'E']
['E' 'B' 'D' 'C' 'A']
['C' 'D' 'B' 'E' 'A']
['C' 'E' 'A' 'B' 'D']
['A' 'E' 'B' 'D' 'C']
['E' 'B' 'D' 'A' 'C']
['E' 'C' 'B' 'D' 'A']
['C' 'D' 'E' 'B' 'A']
['E' 'A' 'D' 'B' 'C']
['E' 'D' 'B' 'A' 'C']
['E' 'C' 'B' 'A' 'D']
['D' 'E' 'B' 'C' 'A']
['A' 'B' 'D' 'E' 'C']
['A' 'E' 'B' 'D' 'C']
['E' 'A' 'D' 'B' 'C']
['A' 'E' 'B' 'D' 'C']
['D' 'E' 'B' 'C' 'A']
['A' 'C' 'D' 'E' 'B']
['E' 'C' 'D' 'B' 'A']
['C' 'D' 'E' 'A' 'B']
['E' 'A' 'B' 'C' 'D']
['E' 'A' 'D' 'C' 'B']
['D' 'A' 'E' 'C' 'B']
['C' 'E' 'A' 'B' 'D']
['D' 'A' 'B' 'C' 'E']
['E' 'B' 'A' 'C' 'D']
['E' 'A' 'B' 'C' 'D']
['E' 'A' 'D' 'B' 'C']
['E' 'B' 'D' 'C' 'A']
['B' 'A' 'D' 'C' 'E']
['C' 'D' 'A' 'E' 'B']
['E' 'D' 'B' 'A' 'C']
['C' 'E' 'A' 'B' 'D']]

```

[6]: *#Definimos una funcion que agarrara los dos progenitores y producira una*
↳ descendencia

```

def cruce_progenitores(lista_progenitores_a, lista_progenitores_b):
    #seleccionamos la descedencia
    descendencia = lista_progenitores_a[0:5]

```

```

#iteramos sobre la otra lista de progenitores
for nodo in lista_progenitores_b:
    #si no esta en la descendencia
    if not nodo in descendencia:
        #concatenamos la nueva descendencia
        descendencia = np.concatenate((descendencia, [nodo]))
    #retornamos la descendencia
    return descendencia

#definimos la funcion para crear una nueva poblacion de soluciones
def crear_poblacion(lista_progenitores):
    #array de la nueva poblacion
    nueva_poblacion = []
    #iteramos sobre la lista de progenitores
    for i in range(lista_progenitores.shape[1]):
        #agarramos los dos progenitores
        progenitor_a, progenitor_b = lista_progenitores[0][i], lista_progenitores[1][i]
        #encontramos la descendencia
        descendencia = cruce_progenitores(progenitor_a, progenitor_b)
        #añadimos al array
        nueva_poblacion.append(descendencia)
    #retornamos la nueva poblacion
    return np.array(nueva_poblacion)

#obtenemos la nueva poblacion
nueva_poblacion = crear_poblacion(lista_progenitores)
print(nueva_poblacion)

```

```

[['A' 'E' 'B' 'D' 'C']
 ['B' 'A' 'E' 'D' 'C']
 ['E' 'A' 'D' 'B' 'C']
 ['E' 'A' 'D' 'B' 'C']
 ['D' 'B' 'C' 'E' 'A']
 ['E' 'B' 'A' 'D' 'C']
 ['A' 'E' 'B' 'D' 'C']
 ['C' 'B' 'D' 'E' 'A']
 ['C' 'D' 'E' 'A' 'B']
 ['C' 'D' 'B' 'E' 'A']
 ['E' 'A' 'B' 'C' 'D']
 ['E' 'C' 'B' 'D' 'A']
 ['C' 'E' 'A' 'B' 'D']
 ['D' 'E' 'B' 'C' 'A']
 ['D' 'E' 'B' 'C' 'A']
 ['C' 'B' 'D' 'E' 'A']
 ['E' 'C' 'B' 'A' 'D']
 ['E' 'C' 'D' 'B' 'A']
 ['E' 'B' 'D' 'A' 'C']

```

```

['E' 'A' 'D' 'B' 'C']
['E' 'A' 'D' 'B' 'C']
['D' 'B' 'C' 'E' 'A']
['C' 'A' 'B' 'D' 'E']
['E' 'C' 'B' 'D' 'A']
['A' 'B' 'C' 'E' 'D']
['E' 'A' 'D' 'B' 'C']
['D' 'B' 'C' 'E' 'A']
['C' 'B' 'D' 'A' 'E']
['E' 'B' 'D' 'C' 'A']
['A' 'C' 'D' 'E' 'B']
['C' 'A' 'B' 'D' 'E']
['B' 'A' 'E' 'D' 'C']
['C' 'B' 'D' 'A' 'E']
['C' 'A' 'B' 'D' 'E']
['A' 'C' 'D' 'E' 'B']
['C' 'D' 'E' 'A' 'B']
['D' 'B' 'A' 'E' 'C']
['D' 'E' 'B' 'C' 'A']
['D' 'A' 'E' 'C' 'B']
['C' 'D' 'A' 'E' 'B']
['E' 'B' 'D' 'C' 'A']
['D' 'B' 'A' 'E' 'C']
['A' 'E' 'B' 'D' 'C']
['E' 'C' 'D' 'B' 'A']
['B' 'A' 'E' 'D' 'C']
['B' 'E' 'C' 'D' 'A']
['E' 'A' 'D' 'C' 'B']
['E' 'B' 'A' 'C' 'D']
['D' 'B' 'A' 'E' 'C']
['C' 'D' 'E' 'A' 'B']

```

```

[7]: #Definiremos una funcion para la mutacion de la descendencia
def mutar_descendencia(descendencia):
    #iteramos sobre el producto de la cantidad de nodos y la probabilidad de
    ↪mutacion
    for i in range(int(n_nodos * probabilidad_mutacion)):
        #escogemos un numero aleatorio entre 0 y el numero de nodos
        a = np.random.randint(0, n_nodos)
        #escogemos un numero aleatorio entre 0 y el numero de nodos
        b = np.random.randint(0, n_nodos)
        #realizamos la mutacion con los indices anteriores
        descendencia[a], descendencia[b] = descendencia[b], descendencia[a]
    #retornamos la mutacion realizada
    return descendencia

```



```

#Definimos la funcion para realizar la mutacion en todo el conjunto de
↳soluciones
def mutar_poblacion(nueva_poblacion):
    #array de la poblacion mutada
    poblacion_mutada = []
    #iteramos sobre cada descendencia de la nueva poblacion
    for descendencia in (nueva_poblacion):
        #añadimos la poblacion mutada a un array
        poblacion_mutada.append(mutar_descendencia(descendencia))
    #retornamos el array de poblacion mutada
    return np.array(poblacion_mutada)

#Obtenemos la poblacion mutada
poblacion_mutada = mutar_poblacion(nueva_poblacion)
print(poblacion_mutada)

```

```

[['A' 'E' 'B' 'C' 'D']
 ['B' 'D' 'E' 'A' 'C']
 ['E' 'A' 'D' 'B' 'C']
 ['B' 'A' 'D' 'E' 'C']
 ['D' 'B' 'C' 'E' 'A']
 ['E' 'B' 'A' 'D' 'C']
 ['A' 'D' 'B' 'E' 'C']
 ['C' 'B' 'D' 'E' 'A']
 ['B' 'D' 'E' 'A' 'C']
 ['B' 'D' 'C' 'E' 'A']
 ['E' 'A' 'D' 'C' 'B']
 ['E' 'C' 'A' 'D' 'B']
 ['A' 'E' 'C' 'B' 'D']
 ['B' 'E' 'D' 'C' 'A']
 ['D' 'A' 'B' 'C' 'E']
 ['C' 'B' 'E' 'D' 'A']
 ['E' 'B' 'C' 'A' 'D']
 ['D' 'C' 'E' 'B' 'A']
 ['E' 'B' 'A' 'D' 'C']
 ['E' 'A' 'D' 'B' 'C']
 ['C' 'A' 'D' 'B' 'E']
 ['B' 'D' 'C' 'E' 'A']
 ['C' 'A' 'D' 'B' 'E']
 ['E' 'D' 'B' 'C' 'A']
 ['A' 'B' 'C' 'E' 'D']
 ['C' 'A' 'D' 'B' 'E']
 ['D' 'B' 'C' 'E' 'A']
 ['C' 'D' 'B' 'A' 'E']
 ['E' 'C' 'D' 'B' 'A']
 ['A' 'C' 'D' 'B' 'E']
 ['B' 'A' 'C' 'D' 'E']
 ['B' 'A' 'C' 'D' 'E']

```

```

['A' 'B' 'D' 'C' 'E']
['B' 'A' 'C' 'D' 'E']
['A' 'C' 'E' 'D' 'B']
['C' 'A' 'E' 'D' 'B']
['D' 'B' 'E' 'A' 'C']
['D' 'E' 'B' 'A' 'C']
['D' 'A' 'B' 'C' 'E']
['C' 'B' 'A' 'E' 'D']
['E' 'B' 'D' 'A' 'C']
['D' 'C' 'A' 'E' 'B']
['B' 'E' 'A' 'D' 'C']
['E' 'C' 'D' 'B' 'A']
['B' 'C' 'E' 'D' 'A']
['E' 'B' 'C' 'D' 'A']
['E' 'D' 'A' 'C' 'B']
['E' 'B' 'A' 'C' 'D']
['D' 'B' 'A' 'E' 'C']
['C' 'E' 'D' 'A' 'B']]

```

```

[8]: #Definimos una funcion para registrar una generacion en el archivo csv
def registrar_generacion(generacion, costo_soluciones, poblacion_mutada,
    ↪costo_minimo, registros):
    #definimos un sw para la mejor solucion
    sw = "NO"
    #iteramos sobre el array de costos
    for i in range(len(costo_soluciones)):
        #si tiene el mejor costo cambiamos a si esto para generar el csv
        if (costo_soluciones[i] == costo_minimo):
            sw = "SI"
        else:
            sw = "NO"
        #añadimos un registro donde se almacenara datos soobre la generacion esto
        ↪para generar el csv
        registro = [generacion, poblacion_mutada[i][0], poblacion_mutada[i][1],
    ↪poblacion_mutada[i][2], poblacion_mutada[i][3], poblacion_mutada[i][4],
    ↪costo_soluciones[i], sw]
        #lo almacenamos en un array
        registros.append(registro)
    #devolvemos los registros obtenidos
    return registros

```

```

[9]: #Obtendremos la mejor solucion
#En el array mejor solucion almacenaremos tres diferentes datos
#el primero hara referencia el numero de iteracion
#el segundo al costo de la solucion y el tercero a la solucion como tal
mejor_solucion = [-1, np.inf, np.array([])]
#Definimos una matriz donde almacenaremos registros de las generaciones

```

```

registros = []
#realizamos 10000 iteraciones
for i in range(1000):
    #Evaluamos el costo de las mutaciones
    costo_soluciones = evaluar_posibles_soluciones(poblacion_mutada,
    ↪diccionario_distancias)
    #comparamos el costo minimo de la poblacion actual con el costo de la mejor
    ↪solucion encontrada
    #Si se encuentra un valor mas optimo se actualiza la mejor solucion
    if costo_soluciones.min() < mejor_solucion[1]:
        #actualizamos el valor de la mejor solucion
        mejor_solucion[0] = i
        mejor_solucion[1] = costo_soluciones.min()
        mejor_solucion[2] = np.array(poblacion_mutada)[costo_soluciones.min() ==
    ↪costo_soluciones]
    if i % 100 == 0:
        registros = registrar_generacion(i+1, costo_soluciones, poblacion_mutada,
    ↪costo_soluciones.min(), registros)
        #mostraremos cada iteracion y su costo
        print(f"Iteracion: {i} Costo: {costo_soluciones.min()}")
        #Realizamos una nueva seleccion de progenitores para esta iteracion
        lista_progenitores = seleccion_progenitor(conjunto_soluciones,
    ↪costo_soluciones)
        #realizamos un cruce para la nueva descendencia
        conjunto_soluciones = crear_poblacion(lista_progenitores)
        #aplicamos la mutacion al nuevo cruce
        poblacion_mutada= mutar_poblacion(conjunto_soluciones)

```

```

Iteracion: 0 Costo: 37.0
Iteracion: 1 Costo: 37.0
Iteracion: 2 Costo: 37.0
Iteracion: 3 Costo: 37.0
Iteracion: 4 Costo: 37.0
Iteracion: 5 Costo: 37.0
Iteracion: 6 Costo: 37.0
Iteracion: 7 Costo: 37.0
Iteracion: 8 Costo: 37.0
Iteracion: 9 Costo: 37.0
Iteracion: 10 Costo: 37.0
Iteracion: 11 Costo: 37.0
Iteracion: 12 Costo: 37.0
Iteracion: 13 Costo: 37.0
Iteracion: 14 Costo: 37.0
Iteracion: 15 Costo: 37.0
Iteracion: 16 Costo: 37.0
Iteracion: 17 Costo: 37.0
Iteracion: 18 Costo: 37.0

```

Iteracion: 19 Costo: 37.0
Iteracion: 20 Costo: 37.0
Iteracion: 21 Costo: 37.0
Iteracion: 22 Costo: 37.0
Iteracion: 23 Costo: 37.0
Iteracion: 24 Costo: 37.0
Iteracion: 25 Costo: 37.0
Iteracion: 26 Costo: 37.0
Iteracion: 27 Costo: 37.0
Iteracion: 28 Costo: 37.0
Iteracion: 29 Costo: 37.0
Iteracion: 30 Costo: 37.0
Iteracion: 31 Costo: 37.0
Iteracion: 32 Costo: 37.0
Iteracion: 33 Costo: 37.0
Iteracion: 34 Costo: 37.0
Iteracion: 35 Costo: 37.0
Iteracion: 36 Costo: 37.0
Iteracion: 37 Costo: 37.0
Iteracion: 38 Costo: 37.0
Iteracion: 39 Costo: 37.0
Iteracion: 40 Costo: 37.0
Iteracion: 41 Costo: 37.0
Iteracion: 42 Costo: 37.0
Iteracion: 43 Costo: 37.0
Iteracion: 44 Costo: 37.0
Iteracion: 45 Costo: 37.0
Iteracion: 46 Costo: 37.0
Iteracion: 47 Costo: 37.0
Iteracion: 48 Costo: 37.0
Iteracion: 49 Costo: 37.0
Iteracion: 50 Costo: 37.0
Iteracion: 51 Costo: 37.0
Iteracion: 52 Costo: 37.0
Iteracion: 53 Costo: 37.0
Iteracion: 54 Costo: 37.0
Iteracion: 55 Costo: 37.0
Iteracion: 56 Costo: 37.0
Iteracion: 57 Costo: 37.0
Iteracion: 58 Costo: 37.0
Iteracion: 59 Costo: 37.0
Iteracion: 60 Costo: 37.0
Iteracion: 61 Costo: 37.0
Iteracion: 62 Costo: 37.0
Iteracion: 63 Costo: 37.0
Iteracion: 64 Costo: 37.0
Iteracion: 65 Costo: 37.0
Iteracion: 66 Costo: 37.0

Iteracion: 67 Costo: 37.0
Iteracion: 68 Costo: 42.0
Iteracion: 69 Costo: 37.0
Iteracion: 70 Costo: 37.0
Iteracion: 71 Costo: 37.0
Iteracion: 72 Costo: 37.0
Iteracion: 73 Costo: 37.0
Iteracion: 74 Costo: 37.0
Iteracion: 75 Costo: 37.0
Iteracion: 76 Costo: 37.0
Iteracion: 77 Costo: 37.0
Iteracion: 78 Costo: 37.0
Iteracion: 79 Costo: 37.0
Iteracion: 80 Costo: 37.0
Iteracion: 81 Costo: 37.0
Iteracion: 82 Costo: 37.0
Iteracion: 83 Costo: 37.0
Iteracion: 84 Costo: 37.0
Iteracion: 85 Costo: 37.0
Iteracion: 86 Costo: 37.0
Iteracion: 87 Costo: 37.0
Iteracion: 88 Costo: 37.0
Iteracion: 89 Costo: 37.0
Iteracion: 90 Costo: 37.0
Iteracion: 91 Costo: 37.0
Iteracion: 92 Costo: 37.0
Iteracion: 93 Costo: 37.0
Iteracion: 94 Costo: 37.0
Iteracion: 95 Costo: 37.0
Iteracion: 96 Costo: 37.0
Iteracion: 97 Costo: 37.0
Iteracion: 98 Costo: 37.0
Iteracion: 99 Costo: 37.0
Iteracion: 100 Costo: 37.0
Iteracion: 101 Costo: 37.0
Iteracion: 102 Costo: 37.0
Iteracion: 103 Costo: 37.0
Iteracion: 104 Costo: 37.0
Iteracion: 105 Costo: 37.0
Iteracion: 106 Costo: 37.0
Iteracion: 107 Costo: 37.0
Iteracion: 108 Costo: 37.0
Iteracion: 109 Costo: 37.0
Iteracion: 110 Costo: 37.0
Iteracion: 111 Costo: 37.0
Iteracion: 112 Costo: 37.0
Iteracion: 113 Costo: 37.0
Iteracion: 114 Costo: 37.0

Iteracion: 115 Costo: 37.0
Iteracion: 116 Costo: 37.0
Iteracion: 117 Costo: 37.0
Iteracion: 118 Costo: 37.0
Iteracion: 119 Costo: 37.0
Iteracion: 120 Costo: 37.0
Iteracion: 121 Costo: 37.0
Iteracion: 122 Costo: 37.0
Iteracion: 123 Costo: 37.0
Iteracion: 124 Costo: 37.0
Iteracion: 125 Costo: 37.0
Iteracion: 126 Costo: 37.0
Iteracion: 127 Costo: 37.0
Iteracion: 128 Costo: 37.0
Iteracion: 129 Costo: 37.0
Iteracion: 130 Costo: 37.0
Iteracion: 131 Costo: 37.0
Iteracion: 132 Costo: 37.0
Iteracion: 133 Costo: 37.0
Iteracion: 134 Costo: 37.0
Iteracion: 135 Costo: 37.0
Iteracion: 136 Costo: 37.0
Iteracion: 137 Costo: 37.0
Iteracion: 138 Costo: 37.0
Iteracion: 139 Costo: 37.0
Iteracion: 140 Costo: 37.0
Iteracion: 141 Costo: 37.0
Iteracion: 142 Costo: 37.0
Iteracion: 143 Costo: 37.0
Iteracion: 144 Costo: 37.0
Iteracion: 145 Costo: 37.0
Iteracion: 146 Costo: 37.0
Iteracion: 147 Costo: 37.0
Iteracion: 148 Costo: 37.0
Iteracion: 149 Costo: 37.0
Iteracion: 150 Costo: 37.0
Iteracion: 151 Costo: 42.0
Iteracion: 152 Costo: 37.0
Iteracion: 153 Costo: 42.0
Iteracion: 154 Costo: 37.0
Iteracion: 155 Costo: 37.0
Iteracion: 156 Costo: 37.0
Iteracion: 157 Costo: 37.0
Iteracion: 158 Costo: 37.0
Iteracion: 159 Costo: 42.0
Iteracion: 160 Costo: 37.0
Iteracion: 161 Costo: 37.0
Iteracion: 162 Costo: 37.0

Iteracion: 163 Costo: 37.0
Iteracion: 164 Costo: 37.0
Iteracion: 165 Costo: 37.0
Iteracion: 166 Costo: 37.0
Iteracion: 167 Costo: 37.0
Iteracion: 168 Costo: 37.0
Iteracion: 169 Costo: 37.0
Iteracion: 170 Costo: 37.0
Iteracion: 171 Costo: 37.0
Iteracion: 172 Costo: 37.0
Iteracion: 173 Costo: 37.0
Iteracion: 174 Costo: 37.0
Iteracion: 175 Costo: 37.0
Iteracion: 176 Costo: 37.0
Iteracion: 177 Costo: 37.0
Iteracion: 178 Costo: 37.0
Iteracion: 179 Costo: 37.0
Iteracion: 180 Costo: 37.0
Iteracion: 181 Costo: 37.0
Iteracion: 182 Costo: 37.0
Iteracion: 183 Costo: 37.0
Iteracion: 184 Costo: 37.0
Iteracion: 185 Costo: 37.0
Iteracion: 186 Costo: 37.0
Iteracion: 187 Costo: 37.0
Iteracion: 188 Costo: 37.0
Iteracion: 189 Costo: 37.0
Iteracion: 190 Costo: 37.0
Iteracion: 191 Costo: 37.0
Iteracion: 192 Costo: 37.0
Iteracion: 193 Costo: 37.0
Iteracion: 194 Costo: 42.0
Iteracion: 195 Costo: 37.0
Iteracion: 196 Costo: 37.0
Iteracion: 197 Costo: 37.0
Iteracion: 198 Costo: 37.0
Iteracion: 199 Costo: 37.0
Iteracion: 200 Costo: 37.0
Iteracion: 201 Costo: 37.0
Iteracion: 202 Costo: 37.0
Iteracion: 203 Costo: 37.0
Iteracion: 204 Costo: 37.0
Iteracion: 205 Costo: 37.0
Iteracion: 206 Costo: 37.0
Iteracion: 207 Costo: 37.0
Iteracion: 208 Costo: 37.0
Iteracion: 209 Costo: 37.0
Iteracion: 210 Costo: 37.0

Iteracion: 211 Costo: 37.0
Iteracion: 212 Costo: 37.0
Iteracion: 213 Costo: 37.0
Iteracion: 214 Costo: 37.0
Iteracion: 215 Costo: 37.0
Iteracion: 216 Costo: 37.0
Iteracion: 217 Costo: 37.0
Iteracion: 218 Costo: 37.0
Iteracion: 219 Costo: 37.0
Iteracion: 220 Costo: 37.0
Iteracion: 221 Costo: 37.0
Iteracion: 222 Costo: 37.0
Iteracion: 223 Costo: 37.0
Iteracion: 224 Costo: 37.0
Iteracion: 225 Costo: 37.0
Iteracion: 226 Costo: 42.0
Iteracion: 227 Costo: 37.0
Iteracion: 228 Costo: 37.0
Iteracion: 229 Costo: 37.0
Iteracion: 230 Costo: 37.0
Iteracion: 231 Costo: 37.0
Iteracion: 232 Costo: 37.0
Iteracion: 233 Costo: 37.0
Iteracion: 234 Costo: 37.0
Iteracion: 235 Costo: 37.0
Iteracion: 236 Costo: 37.0
Iteracion: 237 Costo: 37.0
Iteracion: 238 Costo: 37.0
Iteracion: 239 Costo: 37.0
Iteracion: 240 Costo: 37.0
Iteracion: 241 Costo: 37.0
Iteracion: 242 Costo: 37.0
Iteracion: 243 Costo: 37.0
Iteracion: 244 Costo: 37.0
Iteracion: 245 Costo: 37.0
Iteracion: 246 Costo: 37.0
Iteracion: 247 Costo: 37.0
Iteracion: 248 Costo: 37.0
Iteracion: 249 Costo: 37.0
Iteracion: 250 Costo: 37.0
Iteracion: 251 Costo: 37.0
Iteracion: 252 Costo: 37.0
Iteracion: 253 Costo: 37.0
Iteracion: 254 Costo: 37.0
Iteracion: 255 Costo: 37.0
Iteracion: 256 Costo: 37.0
Iteracion: 257 Costo: 37.0
Iteracion: 258 Costo: 37.0

Iteracion: 259 Costo: 37.0
Iteracion: 260 Costo: 37.0
Iteracion: 261 Costo: 37.0
Iteracion: 262 Costo: 37.0
Iteracion: 263 Costo: 37.0
Iteracion: 264 Costo: 37.0
Iteracion: 265 Costo: 37.0
Iteracion: 266 Costo: 37.0
Iteracion: 267 Costo: 37.0
Iteracion: 268 Costo: 37.0
Iteracion: 269 Costo: 42.0
Iteracion: 270 Costo: 37.0
Iteracion: 271 Costo: 37.0
Iteracion: 272 Costo: 37.0
Iteracion: 273 Costo: 37.0
Iteracion: 274 Costo: 37.0
Iteracion: 275 Costo: 37.0
Iteracion: 276 Costo: 37.0
Iteracion: 277 Costo: 37.0
Iteracion: 278 Costo: 37.0
Iteracion: 279 Costo: 37.0
Iteracion: 280 Costo: 37.0
Iteracion: 281 Costo: 37.0
Iteracion: 282 Costo: 37.0
Iteracion: 283 Costo: 37.0
Iteracion: 284 Costo: 37.0
Iteracion: 285 Costo: 37.0
Iteracion: 286 Costo: 37.0
Iteracion: 287 Costo: 37.0
Iteracion: 288 Costo: 37.0
Iteracion: 289 Costo: 37.0
Iteracion: 290 Costo: 37.0
Iteracion: 291 Costo: 37.0
Iteracion: 292 Costo: 37.0
Iteracion: 293 Costo: 37.0
Iteracion: 294 Costo: 37.0
Iteracion: 295 Costo: 37.0
Iteracion: 296 Costo: 37.0
Iteracion: 297 Costo: 37.0
Iteracion: 298 Costo: 37.0
Iteracion: 299 Costo: 37.0
Iteracion: 300 Costo: 37.0
Iteracion: 301 Costo: 37.0
Iteracion: 302 Costo: 37.0
Iteracion: 303 Costo: 37.0
Iteracion: 304 Costo: 37.0
Iteracion: 305 Costo: 37.0
Iteracion: 306 Costo: 37.0

Iteracion: 307 Costo: 37.0
Iteracion: 308 Costo: 37.0
Iteracion: 309 Costo: 37.0
Iteracion: 310 Costo: 37.0
Iteracion: 311 Costo: 37.0
Iteracion: 312 Costo: 37.0
Iteracion: 313 Costo: 37.0
Iteracion: 314 Costo: 37.0
Iteracion: 315 Costo: 37.0
Iteracion: 316 Costo: 37.0
Iteracion: 317 Costo: 37.0
Iteracion: 318 Costo: 42.0
Iteracion: 319 Costo: 37.0
Iteracion: 320 Costo: 37.0
Iteracion: 321 Costo: 37.0
Iteracion: 322 Costo: 37.0
Iteracion: 323 Costo: 37.0
Iteracion: 324 Costo: 37.0
Iteracion: 325 Costo: 37.0
Iteracion: 326 Costo: 37.0
Iteracion: 327 Costo: 37.0
Iteracion: 328 Costo: 37.0
Iteracion: 329 Costo: 37.0
Iteracion: 330 Costo: 37.0
Iteracion: 331 Costo: 37.0
Iteracion: 332 Costo: 37.0
Iteracion: 333 Costo: 42.0
Iteracion: 334 Costo: 37.0
Iteracion: 335 Costo: 37.0
Iteracion: 336 Costo: 37.0
Iteracion: 337 Costo: 37.0
Iteracion: 338 Costo: 37.0
Iteracion: 339 Costo: 37.0
Iteracion: 340 Costo: 37.0
Iteracion: 341 Costo: 37.0
Iteracion: 342 Costo: 37.0
Iteracion: 343 Costo: 37.0
Iteracion: 344 Costo: 37.0
Iteracion: 345 Costo: 37.0
Iteracion: 346 Costo: 37.0
Iteracion: 347 Costo: 37.0
Iteracion: 348 Costo: 37.0
Iteracion: 349 Costo: 37.0
Iteracion: 350 Costo: 37.0
Iteracion: 351 Costo: 37.0
Iteracion: 352 Costo: 37.0
Iteracion: 353 Costo: 37.0
Iteracion: 354 Costo: 37.0

Iteracion: 355 Costo: 37.0
Iteracion: 356 Costo: 37.0
Iteracion: 357 Costo: 37.0
Iteracion: 358 Costo: 37.0
Iteracion: 359 Costo: 37.0
Iteracion: 360 Costo: 37.0
Iteracion: 361 Costo: 37.0
Iteracion: 362 Costo: 37.0
Iteracion: 363 Costo: 37.0
Iteracion: 364 Costo: 37.0
Iteracion: 365 Costo: 37.0
Iteracion: 366 Costo: 37.0
Iteracion: 367 Costo: 37.0
Iteracion: 368 Costo: 37.0
Iteracion: 369 Costo: 37.0
Iteracion: 370 Costo: 37.0
Iteracion: 371 Costo: 37.0
Iteracion: 372 Costo: 37.0
Iteracion: 373 Costo: 37.0
Iteracion: 374 Costo: 37.0
Iteracion: 375 Costo: 37.0
Iteracion: 376 Costo: 37.0
Iteracion: 377 Costo: 37.0
Iteracion: 378 Costo: 37.0
Iteracion: 379 Costo: 37.0
Iteracion: 380 Costo: 37.0
Iteracion: 381 Costo: 37.0
Iteracion: 382 Costo: 37.0
Iteracion: 383 Costo: 37.0
Iteracion: 384 Costo: 37.0
Iteracion: 385 Costo: 42.0
Iteracion: 386 Costo: 37.0
Iteracion: 387 Costo: 37.0
Iteracion: 388 Costo: 37.0
Iteracion: 389 Costo: 37.0
Iteracion: 390 Costo: 37.0
Iteracion: 391 Costo: 37.0
Iteracion: 392 Costo: 37.0
Iteracion: 393 Costo: 37.0
Iteracion: 394 Costo: 37.0
Iteracion: 395 Costo: 37.0
Iteracion: 396 Costo: 37.0
Iteracion: 397 Costo: 37.0
Iteracion: 398 Costo: 37.0
Iteracion: 399 Costo: 37.0
Iteracion: 400 Costo: 37.0
Iteracion: 401 Costo: 37.0
Iteracion: 402 Costo: 37.0

Iteracion: 403 Costo: 37.0
Iteracion: 404 Costo: 37.0
Iteracion: 405 Costo: 37.0
Iteracion: 406 Costo: 37.0
Iteracion: 407 Costo: 37.0
Iteracion: 408 Costo: 37.0
Iteracion: 409 Costo: 37.0
Iteracion: 410 Costo: 37.0
Iteracion: 411 Costo: 37.0
Iteracion: 412 Costo: 37.0
Iteracion: 413 Costo: 37.0
Iteracion: 414 Costo: 37.0
Iteracion: 415 Costo: 37.0
Iteracion: 416 Costo: 37.0
Iteracion: 417 Costo: 37.0
Iteracion: 418 Costo: 37.0
Iteracion: 419 Costo: 37.0
Iteracion: 420 Costo: 37.0
Iteracion: 421 Costo: 37.0
Iteracion: 422 Costo: 37.0
Iteracion: 423 Costo: 37.0
Iteracion: 424 Costo: 42.0
Iteracion: 425 Costo: 37.0
Iteracion: 426 Costo: 37.0
Iteracion: 427 Costo: 37.0
Iteracion: 428 Costo: 37.0
Iteracion: 429 Costo: 37.0
Iteracion: 430 Costo: 37.0
Iteracion: 431 Costo: 37.0
Iteracion: 432 Costo: 37.0
Iteracion: 433 Costo: 37.0
Iteracion: 434 Costo: 37.0
Iteracion: 435 Costo: 37.0
Iteracion: 436 Costo: 37.0
Iteracion: 437 Costo: 37.0
Iteracion: 438 Costo: 37.0
Iteracion: 439 Costo: 37.0
Iteracion: 440 Costo: 37.0
Iteracion: 441 Costo: 37.0
Iteracion: 442 Costo: 37.0
Iteracion: 443 Costo: 37.0
Iteracion: 444 Costo: 37.0
Iteracion: 445 Costo: 37.0
Iteracion: 446 Costo: 37.0
Iteracion: 447 Costo: 37.0
Iteracion: 448 Costo: 37.0
Iteracion: 449 Costo: 37.0
Iteracion: 450 Costo: 37.0

Iteracion: 451 Costo: 37.0
Iteracion: 452 Costo: 37.0
Iteracion: 453 Costo: 37.0
Iteracion: 454 Costo: 37.0
Iteracion: 455 Costo: 37.0
Iteracion: 456 Costo: 37.0
Iteracion: 457 Costo: 37.0
Iteracion: 458 Costo: 37.0
Iteracion: 459 Costo: 37.0
Iteracion: 460 Costo: 37.0
Iteracion: 461 Costo: 37.0
Iteracion: 462 Costo: 37.0
Iteracion: 463 Costo: 37.0
Iteracion: 464 Costo: 37.0
Iteracion: 465 Costo: 37.0
Iteracion: 466 Costo: 37.0
Iteracion: 467 Costo: 37.0
Iteracion: 468 Costo: 37.0
Iteracion: 469 Costo: 37.0
Iteracion: 470 Costo: 37.0
Iteracion: 471 Costo: 37.0
Iteracion: 472 Costo: 37.0
Iteracion: 473 Costo: 37.0
Iteracion: 474 Costo: 37.0
Iteracion: 475 Costo: 37.0
Iteracion: 476 Costo: 37.0
Iteracion: 477 Costo: 37.0
Iteracion: 478 Costo: 37.0
Iteracion: 479 Costo: 37.0
Iteracion: 480 Costo: 37.0
Iteracion: 481 Costo: 37.0
Iteracion: 482 Costo: 37.0
Iteracion: 483 Costo: 37.0
Iteracion: 484 Costo: 37.0
Iteracion: 485 Costo: 37.0
Iteracion: 486 Costo: 37.0
Iteracion: 487 Costo: 37.0
Iteracion: 488 Costo: 37.0
Iteracion: 489 Costo: 37.0
Iteracion: 490 Costo: 37.0
Iteracion: 491 Costo: 37.0
Iteracion: 492 Costo: 37.0
Iteracion: 493 Costo: 37.0
Iteracion: 494 Costo: 37.0
Iteracion: 495 Costo: 37.0
Iteracion: 496 Costo: 37.0
Iteracion: 497 Costo: 37.0
Iteracion: 498 Costo: 37.0

Iteracion: 499 Costo: 37.0
Iteracion: 500 Costo: 37.0
Iteracion: 501 Costo: 37.0
Iteracion: 502 Costo: 37.0
Iteracion: 503 Costo: 37.0
Iteracion: 504 Costo: 37.0
Iteracion: 505 Costo: 37.0
Iteracion: 506 Costo: 37.0
Iteracion: 507 Costo: 37.0
Iteracion: 508 Costo: 37.0
Iteracion: 509 Costo: 37.0
Iteracion: 510 Costo: 37.0
Iteracion: 511 Costo: 37.0
Iteracion: 512 Costo: 37.0
Iteracion: 513 Costo: 37.0
Iteracion: 514 Costo: 37.0
Iteracion: 515 Costo: 37.0
Iteracion: 516 Costo: 37.0
Iteracion: 517 Costo: 37.0
Iteracion: 518 Costo: 37.0
Iteracion: 519 Costo: 37.0
Iteracion: 520 Costo: 37.0
Iteracion: 521 Costo: 37.0
Iteracion: 522 Costo: 37.0
Iteracion: 523 Costo: 37.0
Iteracion: 524 Costo: 37.0
Iteracion: 525 Costo: 37.0
Iteracion: 526 Costo: 37.0
Iteracion: 527 Costo: 37.0
Iteracion: 528 Costo: 37.0
Iteracion: 529 Costo: 37.0
Iteracion: 530 Costo: 37.0
Iteracion: 531 Costo: 37.0
Iteracion: 532 Costo: 37.0
Iteracion: 533 Costo: 37.0
Iteracion: 534 Costo: 37.0
Iteracion: 535 Costo: 37.0
Iteracion: 536 Costo: 37.0
Iteracion: 537 Costo: 37.0
Iteracion: 538 Costo: 42.0
Iteracion: 539 Costo: 37.0
Iteracion: 540 Costo: 37.0
Iteracion: 541 Costo: 37.0
Iteracion: 542 Costo: 37.0
Iteracion: 543 Costo: 37.0
Iteracion: 544 Costo: 37.0
Iteracion: 545 Costo: 37.0
Iteracion: 546 Costo: 37.0

Iteracion: 547 Costo: 37.0
Iteracion: 548 Costo: 37.0
Iteracion: 549 Costo: 37.0
Iteracion: 550 Costo: 37.0
Iteracion: 551 Costo: 37.0
Iteracion: 552 Costo: 37.0
Iteracion: 553 Costo: 37.0
Iteracion: 554 Costo: 37.0
Iteracion: 555 Costo: 37.0
Iteracion: 556 Costo: 37.0
Iteracion: 557 Costo: 37.0
Iteracion: 558 Costo: 37.0
Iteracion: 559 Costo: 37.0
Iteracion: 560 Costo: 37.0
Iteracion: 561 Costo: 37.0
Iteracion: 562 Costo: 37.0
Iteracion: 563 Costo: 37.0
Iteracion: 564 Costo: 37.0
Iteracion: 565 Costo: 37.0
Iteracion: 566 Costo: 37.0
Iteracion: 567 Costo: 37.0
Iteracion: 568 Costo: 37.0
Iteracion: 569 Costo: 37.0
Iteracion: 570 Costo: 37.0
Iteracion: 571 Costo: 37.0
Iteracion: 572 Costo: 37.0
Iteracion: 573 Costo: 37.0
Iteracion: 574 Costo: 42.0
Iteracion: 575 Costo: 37.0
Iteracion: 576 Costo: 37.0
Iteracion: 577 Costo: 37.0
Iteracion: 578 Costo: 37.0
Iteracion: 579 Costo: 37.0
Iteracion: 580 Costo: 37.0
Iteracion: 581 Costo: 37.0
Iteracion: 582 Costo: 37.0
Iteracion: 583 Costo: 37.0
Iteracion: 584 Costo: 37.0
Iteracion: 585 Costo: 37.0
Iteracion: 586 Costo: 37.0
Iteracion: 587 Costo: 37.0
Iteracion: 588 Costo: 37.0
Iteracion: 589 Costo: 37.0
Iteracion: 590 Costo: 37.0
Iteracion: 591 Costo: 37.0
Iteracion: 592 Costo: 37.0
Iteracion: 593 Costo: 37.0
Iteracion: 594 Costo: 37.0

Iteracion: 595 Costo: 37.0
Iteracion: 596 Costo: 37.0
Iteracion: 597 Costo: 37.0
Iteracion: 598 Costo: 37.0
Iteracion: 599 Costo: 37.0
Iteracion: 600 Costo: 37.0
Iteracion: 601 Costo: 37.0
Iteracion: 602 Costo: 37.0
Iteracion: 603 Costo: 37.0
Iteracion: 604 Costo: 37.0
Iteracion: 605 Costo: 37.0
Iteracion: 606 Costo: 37.0
Iteracion: 607 Costo: 37.0
Iteracion: 608 Costo: 37.0
Iteracion: 609 Costo: 37.0
Iteracion: 610 Costo: 37.0
Iteracion: 611 Costo: 37.0
Iteracion: 612 Costo: 37.0
Iteracion: 613 Costo: 37.0
Iteracion: 614 Costo: 37.0
Iteracion: 615 Costo: 37.0
Iteracion: 616 Costo: 37.0
Iteracion: 617 Costo: 46.0
Iteracion: 618 Costo: 37.0
Iteracion: 619 Costo: 37.0
Iteracion: 620 Costo: 37.0
Iteracion: 621 Costo: 37.0
Iteracion: 622 Costo: 37.0
Iteracion: 623 Costo: 37.0
Iteracion: 624 Costo: 37.0
Iteracion: 625 Costo: 37.0
Iteracion: 626 Costo: 37.0
Iteracion: 627 Costo: 37.0
Iteracion: 628 Costo: 37.0
Iteracion: 629 Costo: 37.0
Iteracion: 630 Costo: 37.0
Iteracion: 631 Costo: 37.0
Iteracion: 632 Costo: 37.0
Iteracion: 633 Costo: 37.0
Iteracion: 634 Costo: 37.0
Iteracion: 635 Costo: 37.0
Iteracion: 636 Costo: 37.0
Iteracion: 637 Costo: 37.0
Iteracion: 638 Costo: 37.0
Iteracion: 639 Costo: 37.0
Iteracion: 640 Costo: 37.0
Iteracion: 641 Costo: 37.0
Iteracion: 642 Costo: 37.0

Iteracion: 643 Costo: 37.0
Iteracion: 644 Costo: 37.0
Iteracion: 645 Costo: 37.0
Iteracion: 646 Costo: 37.0
Iteracion: 647 Costo: 37.0
Iteracion: 648 Costo: 37.0
Iteracion: 649 Costo: 37.0
Iteracion: 650 Costo: 37.0
Iteracion: 651 Costo: 37.0
Iteracion: 652 Costo: 37.0
Iteracion: 653 Costo: 37.0
Iteracion: 654 Costo: 37.0
Iteracion: 655 Costo: 37.0
Iteracion: 656 Costo: 37.0
Iteracion: 657 Costo: 37.0
Iteracion: 658 Costo: 37.0
Iteracion: 659 Costo: 37.0
Iteracion: 660 Costo: 37.0
Iteracion: 661 Costo: 37.0
Iteracion: 662 Costo: 37.0
Iteracion: 663 Costo: 37.0
Iteracion: 664 Costo: 37.0
Iteracion: 665 Costo: 37.0
Iteracion: 666 Costo: 37.0
Iteracion: 667 Costo: 37.0
Iteracion: 668 Costo: 37.0
Iteracion: 669 Costo: 37.0
Iteracion: 670 Costo: 37.0
Iteracion: 671 Costo: 37.0
Iteracion: 672 Costo: 37.0
Iteracion: 673 Costo: 37.0
Iteracion: 674 Costo: 42.0
Iteracion: 675 Costo: 37.0
Iteracion: 676 Costo: 37.0
Iteracion: 677 Costo: 37.0
Iteracion: 678 Costo: 37.0
Iteracion: 679 Costo: 37.0
Iteracion: 680 Costo: 37.0
Iteracion: 681 Costo: 37.0
Iteracion: 682 Costo: 37.0
Iteracion: 683 Costo: 37.0
Iteracion: 684 Costo: 37.0
Iteracion: 685 Costo: 37.0
Iteracion: 686 Costo: 37.0
Iteracion: 687 Costo: 37.0
Iteracion: 688 Costo: 37.0
Iteracion: 689 Costo: 37.0
Iteracion: 690 Costo: 37.0

Iteracion: 691 Costo: 37.0
Iteracion: 692 Costo: 37.0
Iteracion: 693 Costo: 37.0
Iteracion: 694 Costo: 37.0
Iteracion: 695 Costo: 37.0
Iteracion: 696 Costo: 37.0
Iteracion: 697 Costo: 37.0
Iteracion: 698 Costo: 37.0
Iteracion: 699 Costo: 37.0
Iteracion: 700 Costo: 37.0
Iteracion: 701 Costo: 37.0
Iteracion: 702 Costo: 37.0
Iteracion: 703 Costo: 37.0
Iteracion: 704 Costo: 37.0
Iteracion: 705 Costo: 37.0
Iteracion: 706 Costo: 37.0
Iteracion: 707 Costo: 37.0
Iteracion: 708 Costo: 37.0
Iteracion: 709 Costo: 37.0
Iteracion: 710 Costo: 37.0
Iteracion: 711 Costo: 37.0
Iteracion: 712 Costo: 37.0
Iteracion: 713 Costo: 37.0
Iteracion: 714 Costo: 37.0
Iteracion: 715 Costo: 37.0
Iteracion: 716 Costo: 37.0
Iteracion: 717 Costo: 37.0
Iteracion: 718 Costo: 37.0
Iteracion: 719 Costo: 37.0
Iteracion: 720 Costo: 37.0
Iteracion: 721 Costo: 37.0
Iteracion: 722 Costo: 37.0
Iteracion: 723 Costo: 37.0
Iteracion: 724 Costo: 37.0
Iteracion: 725 Costo: 37.0
Iteracion: 726 Costo: 37.0
Iteracion: 727 Costo: 37.0
Iteracion: 728 Costo: 37.0
Iteracion: 729 Costo: 37.0
Iteracion: 730 Costo: 37.0
Iteracion: 731 Costo: 37.0
Iteracion: 732 Costo: 37.0
Iteracion: 733 Costo: 37.0
Iteracion: 734 Costo: 37.0
Iteracion: 735 Costo: 37.0
Iteracion: 736 Costo: 37.0
Iteracion: 737 Costo: 37.0
Iteracion: 738 Costo: 37.0

Iteracion: 739 Costo: 37.0
Iteracion: 740 Costo: 37.0
Iteracion: 741 Costo: 37.0
Iteracion: 742 Costo: 37.0
Iteracion: 743 Costo: 37.0
Iteracion: 744 Costo: 37.0
Iteracion: 745 Costo: 37.0
Iteracion: 746 Costo: 37.0
Iteracion: 747 Costo: 37.0
Iteracion: 748 Costo: 37.0
Iteracion: 749 Costo: 37.0
Iteracion: 750 Costo: 37.0
Iteracion: 751 Costo: 37.0
Iteracion: 752 Costo: 37.0
Iteracion: 753 Costo: 37.0
Iteracion: 754 Costo: 37.0
Iteracion: 755 Costo: 37.0
Iteracion: 756 Costo: 37.0
Iteracion: 757 Costo: 37.0
Iteracion: 758 Costo: 42.0
Iteracion: 759 Costo: 37.0
Iteracion: 760 Costo: 37.0
Iteracion: 761 Costo: 37.0
Iteracion: 762 Costo: 37.0
Iteracion: 763 Costo: 37.0
Iteracion: 764 Costo: 37.0
Iteracion: 765 Costo: 37.0
Iteracion: 766 Costo: 37.0
Iteracion: 767 Costo: 37.0
Iteracion: 768 Costo: 37.0
Iteracion: 769 Costo: 37.0
Iteracion: 770 Costo: 37.0
Iteracion: 771 Costo: 37.0
Iteracion: 772 Costo: 37.0
Iteracion: 773 Costo: 37.0
Iteracion: 774 Costo: 37.0
Iteracion: 775 Costo: 37.0
Iteracion: 776 Costo: 37.0
Iteracion: 777 Costo: 37.0
Iteracion: 778 Costo: 37.0
Iteracion: 779 Costo: 37.0
Iteracion: 780 Costo: 37.0
Iteracion: 781 Costo: 37.0
Iteracion: 782 Costo: 37.0
Iteracion: 783 Costo: 37.0
Iteracion: 784 Costo: 37.0
Iteracion: 785 Costo: 37.0
Iteracion: 786 Costo: 37.0

Iteracion: 787 Costo: 37.0
Iteracion: 788 Costo: 37.0
Iteracion: 789 Costo: 42.0
Iteracion: 790 Costo: 37.0
Iteracion: 791 Costo: 37.0
Iteracion: 792 Costo: 37.0
Iteracion: 793 Costo: 37.0
Iteracion: 794 Costo: 37.0
Iteracion: 795 Costo: 37.0
Iteracion: 796 Costo: 37.0
Iteracion: 797 Costo: 37.0
Iteracion: 798 Costo: 37.0
Iteracion: 799 Costo: 37.0
Iteracion: 800 Costo: 37.0
Iteracion: 801 Costo: 37.0
Iteracion: 802 Costo: 37.0
Iteracion: 803 Costo: 37.0
Iteracion: 804 Costo: 37.0
Iteracion: 805 Costo: 37.0
Iteracion: 806 Costo: 37.0
Iteracion: 807 Costo: 37.0
Iteracion: 808 Costo: 37.0
Iteracion: 809 Costo: 37.0
Iteracion: 810 Costo: 37.0
Iteracion: 811 Costo: 37.0
Iteracion: 812 Costo: 37.0
Iteracion: 813 Costo: 37.0
Iteracion: 814 Costo: 37.0
Iteracion: 815 Costo: 37.0
Iteracion: 816 Costo: 37.0
Iteracion: 817 Costo: 37.0
Iteracion: 818 Costo: 37.0
Iteracion: 819 Costo: 37.0
Iteracion: 820 Costo: 37.0
Iteracion: 821 Costo: 37.0
Iteracion: 822 Costo: 37.0
Iteracion: 823 Costo: 37.0
Iteracion: 824 Costo: 37.0
Iteracion: 825 Costo: 37.0
Iteracion: 826 Costo: 37.0
Iteracion: 827 Costo: 37.0
Iteracion: 828 Costo: 37.0
Iteracion: 829 Costo: 37.0
Iteracion: 830 Costo: 42.0
Iteracion: 831 Costo: 37.0
Iteracion: 832 Costo: 37.0
Iteracion: 833 Costo: 37.0
Iteracion: 834 Costo: 37.0

Iteracion: 835 Costo: 37.0
Iteracion: 836 Costo: 37.0
Iteracion: 837 Costo: 37.0
Iteracion: 838 Costo: 37.0
Iteracion: 839 Costo: 37.0
Iteracion: 840 Costo: 37.0
Iteracion: 841 Costo: 37.0
Iteracion: 842 Costo: 37.0
Iteracion: 843 Costo: 37.0
Iteracion: 844 Costo: 37.0
Iteracion: 845 Costo: 37.0
Iteracion: 846 Costo: 37.0
Iteracion: 847 Costo: 37.0
Iteracion: 848 Costo: 37.0
Iteracion: 849 Costo: 37.0
Iteracion: 850 Costo: 37.0
Iteracion: 851 Costo: 37.0
Iteracion: 852 Costo: 37.0
Iteracion: 853 Costo: 37.0
Iteracion: 854 Costo: 37.0
Iteracion: 855 Costo: 37.0
Iteracion: 856 Costo: 37.0
Iteracion: 857 Costo: 37.0
Iteracion: 858 Costo: 37.0
Iteracion: 859 Costo: 37.0
Iteracion: 860 Costo: 37.0
Iteracion: 861 Costo: 37.0
Iteracion: 862 Costo: 37.0
Iteracion: 863 Costo: 37.0
Iteracion: 864 Costo: 37.0
Iteracion: 865 Costo: 37.0
Iteracion: 866 Costo: 37.0
Iteracion: 867 Costo: 37.0
Iteracion: 868 Costo: 37.0
Iteracion: 869 Costo: 37.0
Iteracion: 870 Costo: 37.0
Iteracion: 871 Costo: 37.0
Iteracion: 872 Costo: 37.0
Iteracion: 873 Costo: 37.0
Iteracion: 874 Costo: 37.0
Iteracion: 875 Costo: 37.0
Iteracion: 876 Costo: 37.0
Iteracion: 877 Costo: 37.0
Iteracion: 878 Costo: 37.0
Iteracion: 879 Costo: 37.0
Iteracion: 880 Costo: 37.0
Iteracion: 881 Costo: 37.0
Iteracion: 882 Costo: 37.0

Iteracion: 883 Costo: 37.0
Iteracion: 884 Costo: 42.0
Iteracion: 885 Costo: 42.0
Iteracion: 886 Costo: 42.0
Iteracion: 887 Costo: 37.0
Iteracion: 888 Costo: 37.0
Iteracion: 889 Costo: 37.0
Iteracion: 890 Costo: 37.0
Iteracion: 891 Costo: 42.0
Iteracion: 892 Costo: 37.0
Iteracion: 893 Costo: 37.0
Iteracion: 894 Costo: 37.0
Iteracion: 895 Costo: 37.0
Iteracion: 896 Costo: 37.0
Iteracion: 897 Costo: 37.0
Iteracion: 898 Costo: 37.0
Iteracion: 899 Costo: 37.0
Iteracion: 900 Costo: 37.0
Iteracion: 901 Costo: 37.0
Iteracion: 902 Costo: 37.0
Iteracion: 903 Costo: 37.0
Iteracion: 904 Costo: 37.0
Iteracion: 905 Costo: 37.0
Iteracion: 906 Costo: 37.0
Iteracion: 907 Costo: 37.0
Iteracion: 908 Costo: 37.0
Iteracion: 909 Costo: 37.0
Iteracion: 910 Costo: 37.0
Iteracion: 911 Costo: 37.0
Iteracion: 912 Costo: 37.0
Iteracion: 913 Costo: 37.0
Iteracion: 914 Costo: 37.0
Iteracion: 915 Costo: 37.0
Iteracion: 916 Costo: 37.0
Iteracion: 917 Costo: 37.0
Iteracion: 918 Costo: 37.0
Iteracion: 919 Costo: 37.0
Iteracion: 920 Costo: 37.0
Iteracion: 921 Costo: 37.0
Iteracion: 922 Costo: 37.0
Iteracion: 923 Costo: 37.0
Iteracion: 924 Costo: 37.0
Iteracion: 925 Costo: 37.0
Iteracion: 926 Costo: 37.0
Iteracion: 927 Costo: 37.0
Iteracion: 928 Costo: 37.0
Iteracion: 929 Costo: 37.0
Iteracion: 930 Costo: 37.0

Iteracion: 931 Costo: 37.0
Iteracion: 932 Costo: 37.0
Iteracion: 933 Costo: 37.0
Iteracion: 934 Costo: 37.0
Iteracion: 935 Costo: 37.0
Iteracion: 936 Costo: 37.0
Iteracion: 937 Costo: 37.0
Iteracion: 938 Costo: 37.0
Iteracion: 939 Costo: 37.0
Iteracion: 940 Costo: 37.0
Iteracion: 941 Costo: 37.0
Iteracion: 942 Costo: 37.0
Iteracion: 943 Costo: 37.0
Iteracion: 944 Costo: 37.0
Iteracion: 945 Costo: 37.0
Iteracion: 946 Costo: 37.0
Iteracion: 947 Costo: 37.0
Iteracion: 948 Costo: 37.0
Iteracion: 949 Costo: 37.0
Iteracion: 950 Costo: 37.0
Iteracion: 951 Costo: 37.0
Iteracion: 952 Costo: 37.0
Iteracion: 953 Costo: 37.0
Iteracion: 954 Costo: 37.0
Iteracion: 955 Costo: 37.0
Iteracion: 956 Costo: 42.0
Iteracion: 957 Costo: 37.0
Iteracion: 958 Costo: 42.0
Iteracion: 959 Costo: 37.0
Iteracion: 960 Costo: 37.0
Iteracion: 961 Costo: 37.0
Iteracion: 962 Costo: 37.0
Iteracion: 963 Costo: 37.0
Iteracion: 964 Costo: 37.0
Iteracion: 965 Costo: 37.0
Iteracion: 966 Costo: 37.0
Iteracion: 967 Costo: 37.0
Iteracion: 968 Costo: 37.0
Iteracion: 969 Costo: 37.0
Iteracion: 970 Costo: 37.0
Iteracion: 971 Costo: 37.0
Iteracion: 972 Costo: 37.0
Iteracion: 973 Costo: 37.0
Iteracion: 974 Costo: 37.0
Iteracion: 975 Costo: 37.0
Iteracion: 976 Costo: 37.0
Iteracion: 977 Costo: 37.0
Iteracion: 978 Costo: 37.0

```

Iteracion: 979 Costo: 37.0
Iteracion: 980 Costo: 37.0
Iteracion: 981 Costo: 37.0
Iteracion: 982 Costo: 37.0
Iteracion: 983 Costo: 37.0
Iteracion: 984 Costo: 37.0
Iteracion: 985 Costo: 37.0
Iteracion: 986 Costo: 37.0
Iteracion: 987 Costo: 37.0
Iteracion: 988 Costo: 37.0
Iteracion: 989 Costo: 37.0
Iteracion: 990 Costo: 42.0
Iteracion: 991 Costo: 37.0
Iteracion: 992 Costo: 37.0
Iteracion: 993 Costo: 37.0
Iteracion: 994 Costo: 37.0
Iteracion: 995 Costo: 37.0
Iteracion: 996 Costo: 37.0
Iteracion: 997 Costo: 37.0
Iteracion: 998 Costo: 37.0
Iteracion: 999 Costo: 37.0

```

```

[10]: #Mostramos la mejor solucion
print(f"La mejor solucion se encontro en la iteracion numero_
↪{mejor_solucion[0]}, el costo es de {mejor_solucion[1]}, y la ruta optima es_
↪{mejor_solucion[2]}")

```

La mejor solucion se encontro en la iteracion numero 0, el costo es de 37.0, y la ruta optima es [['A' 'D' 'B' 'E' 'C']

```

['E' 'C' 'A' 'D' 'B']
['C' 'A' 'D' 'B' 'E']
['C' 'A' 'D' 'B' 'E']
['C' 'A' 'D' 'B' 'E']
['E' 'B' 'D' 'A' 'C']]

```

```

[11]: #Procederemos a generar el csv
#from google.colab import drive
#importamos la libreria para leer csv
import csv
#mostamos la unidad de drive
#drive.mount('/content/drive')
#asignamos la ruta del archivo
#archivo = '/content/drive/MyDrive/data/INF-354-2P-P3.csv'
archivo = 'C:/Users/TOSHIBA/Desktop/INF - 354/Segundo Parcial/P3/INF-354-2P-P3.
↪csv'
#definimos las cabeceras del archivo csv
cabeceras = ['Generacion', 'Nodo_1', 'Nodo_2', 'Nodo_3', 'Nodo_4', 'Nodo_5',
↪'Costo', 'Mejor_recorrido']

```



```
#abrimos el archivo csv para escribirlo
with open(archivo, 'w', newline='') as f:
    #construimos el escritor de csv
    escritor = csv.writer(f)
    #escribimos las cabeceras
    escritor.writerow(cabeceras)
    #escribimos los registros
    escritor.writerows(registros)
print("Se guardo el csv con exito")
```

Se guardo el csv con exito