|  |  |
| --- | --- |
| escudofi_negro_m2008_jpg | UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  FACULTAD DE INGENIERÍA  ESTRUCTURA DE DATOS Y ALGORITMOS II  M. EN C. ARTURO RODRÍGUEZ GARCÍA  EVALUACIÓN FINAL DE LABORATORIO  TIPO B |

**Nombre de los alumnos:**

**1.- Silva Sandoval Cecilia**

**2.- Silva García Carlos Sebastián**

**3.- Torres Quero Carlos Iván**

**4.- Valderrama Navarro Armando**

*(listar el nombre completo de los integrantes en orden alfabético, empezando por apellido paterno, y borrar esta línea)*

INSTRUCCIONES GENERALES

La evaluación se desarrollará en equipos de 1 a 5 integrantes y será entregado en un plazo de una semana. El formato de entrega será digital en formato pdf y una copia impresa. La revisión se hará directamente sobre el documento digital. Utilizar este documento como plantilla para colocar las respuestas solicitadas. Introduce todas tus respuestas con fuente Corel New tamaño 7.

Enviar por correo electrónico a *arturorogaunam@gmail.com* el archivo pdf de esta evaluación junto con una carpeta titulada Prácticas en donde se incluyan en pdf por separado el reporte de todas las prácticas realizadas en el semestre, con el nombre Practica01.pdf, Practica02.pdf, …, Practica13.pdf. El asunto del correo debe ser EvaluacionFinalGrupoN, donde N es el número de grupo.

Agregar como portada la carátula oficial para entrega de prácticas de laboratorio.

A partir del mapa de estaciones del metro de la Ciudad de México realizar las actividades solicitadas en cada sección.

Las secciones 1, 2 y 3 se trabajarán con lenguaje Python, y la sección 4 con lenguaje C.

Total de aciertos en la evaluación: 40 aciertos



**SECCIÓN 1.- Representación de grafos o árboles (*Valor 10 aciertos)***

1.- En Python, representa en formato de lista el grafo que forman las estaciones del metro de la Ciudad de México, de modo que cada arista sea una lista que contenga dos vértices. Por ejemplo:

Grafo = [ [ ‘hidalgo’ , ‘revolución’] , [ ‘hidalgo’ , ‘guerrero’ ] , [ ‘hidalgo’ , ‘bellasartes’ ] , … ]

Utiliza únicamente letras minúsculas y omite los espacios en blanco para nombrar las estaciones. Por ejemplo, Bellas Artes se representa como ‘bellasartes’. No repitas aristas, es decir, sería incorrecto listar [ ‘hidalgo’ , ‘guerrero’ ] y [ ‘guerrero’ , ‘hidalgo’ ] en forma simultánea.

Muestra aquí tu representación resultante. *(Valor 1 acierto).*

*Grafo=['observatorio', 'tacubaya']*

*['tacubaya', 'juanacatlan']*

*['juanacatlan', 'chapultepec']*

*['chapultepec', 'sevilla']*

*['sevilla', 'insurgentes']*

*['insurgentes', 'cuauhtemoc']*

*['cuauhtemoc', 'balderas']*

*['balderas', 'saltodelagua']*

*['saltodelagua', 'isabellacatolica']*

*['isabellacatolica', 'pinosuarez']*

*['pinosuarez', 'merced']*

*['merced', 'candelaria']*

*['candelaria', 'sanlazaro']*

*['sanlazaro', 'moctezuma']*

*['moctezuma', 'balbuena']*

*['balbuena', 'boulevardpuertoaereo']*

*['boulevardpuertoaereo', 'gomezfarias']*

*['gomezfarias', 'zaragoza']*

*['zaragoza', 'pantitlan']*

*['cuatrocaminos', 'panteones']*

*['panteones', 'tacuba']*

*['tacuba', 'cuitlahuac']*

*['cuitlahuac', 'popotla']*

*['popotla', 'colegiomilitar']*

*['colegiomilitar', 'normal']*

*['normal', 'sancosme']*

*['sancosme', 'revolucion']*

*['revolucion', 'hidalgo']*

*['hidalgo', 'bellasartes']*

*['bellasartes', 'alllende']*

*['allende', 'zocalo']*

*['zocalo', 'pinosuarez']*

*['pinosuarez', 'sanantonioabad']*

*['sanantonioabad', 'chabacano']*

*['chabacano', 'viaducto']*

*['viaducto', 'xola']*

*['xola', 'villadecortes']*

*['villadecortes', 'nativitas']*

*['nativias', 'portales']*

*['portales', 'ermita']*

*['ermita', 'generalanaya']*

*['generalanaya', 'tasquena']*

*['indiosverdes', 'deportivo18demarzo']*

*['deportivo18demarzo', 'potrero']*

*['potrero', 'laraza']*

*['laraza', 'tlatelolco']*

*['tlatelolco', 'guerrero']*

*['guerrero', 'hidalgo']*

*['hidalgo', 'juarez']*

*['juarez', 'balderas']*

*['balderas', 'ninosheroes']*

*['ninosheroes', 'hospitalgeneral']*

*['hospitalgeneral', 'centromedico']*

*['centromedico', 'etiopiaplazadelatransparencia']*

*['etiopiaplazadelatransparencia', 'eugenia']*

*['eugenia', 'divisiondelnorte']*

*['divisiondelnorte', 'zapata']*

*['zapata', 'coyoacan']*

*['coyoacan', 'viverosderechoshumanos']*

*['viverosderechoshumanos', 'miguelangeldequevedo']*

*['miguelangeldequevedo', 'copilco']*

*['copilco', 'universidad']*

*['martincarrera', 'talisman']*

*['talisman', 'bondojito']*

*['bondojito', 'consulado']*

*['consulado', 'canaldelnorte']*

*['canaldelnorte', 'morelos']*

*['morelos', 'candelaria']*

*['candelaria', 'frayservando']*

*['frayservando', 'jamaica']*

*['jamaica', 'santaanita']*

*['politecnico', 'institutodelpetroleo']*

*['institutodelpetroleo', 'autobusesdelnorte']*

*['autobusesdelnorte', 'laraza']*

*['laraza', 'misterios']*

*['misterios', 'vallegomez']*

*['vallegomez', 'consulado']*

*['consulado', 'eduardomolina']*

*['eduardomolina', 'aragon']*

*['aragon', 'oceania']*

*['oceania', 'terminalaerea']*

*['terminalaerea', 'hangares']*

*['hangares', 'pantitlan']*

*['martincarrera', 'lavilla-basilica']*

*['lavilla-basilica', 'deportivo18demarzo']*

*['deportivo18demarzo', 'lindavista']*

*['lindavista', 'institutodelpetroleo']*

*['institutodelpetroleo', 'vallejo']*

*['vallejo', 'norte45']*

*['norte45', 'ferreria']*

*['ferreria', 'azcapotzalco']*

*['azcapotzalco', 'tezozomoc']*

*['tezozomoc', 'elrosario']*

*['lapaz', 'losreyes']*

*['losreyes', 'santamarta']*

*['santamarta', 'acatitla']*

*['acatitla', 'penonviejo']*

*['penonviejo', 'guelatao']*

*['guelatao', 'tepalcates']*

*['tepalcates', 'canaldesanjuan']*

*['canaldesanjuan', 'agricolaoriental']*

*['agricolaoriental.pantitlan']*

*['barrancadelmuerto', 'mixcoac']*

*['mixcoac', 'snantonio']*

*['snantonio', 'snpedrodelospinos']*

*['snpedrodelospinos', 'tacubaya']*

*['tacubaya', 'constituyentes']*

*['constituyentes', 'auditorio']*

*['auditorio', 'polanco']*

*['polanco', 'sanjoaquin']*

*['sanjoaquin', 'tacuba']*

*['tacuba', 'refineria']*

*['refineria', 'camarones']*

*['camarones', 'aquilesserdan']*

*['aquilesserdan', 'elrosario']*

*['constde1917', 'uam1']*

*['uam1', 'cdelaestrella']*

*['cdelaestrella', 'iztapalapa']*

*['iztapalapa', 'atlalilco']*

*['atlalilco', 'escuadron201']*

*['escuadron201', 'aculco']*

*['aculco', 'apatlaco']*

*['apatlaco', 'iztacalco']*

*['iztacalco', 'coyuya']*

*['coyuya', 'santaanita']*

*['santaanita', 'laviga']*

*['laviga', 'chabacano']*

*['chabacano', 'obrera']*

*['obrera', 'doctores']*

*['doctores', 'saltodelagua']*

*['saltodelagua', 'sanjuandeletran']*

*['sanjuandeletran', 'bellasartes']*

*['bellasartes', 'garibaldilagunilla']*

*['tacubaya', 'patriotismo']*

*['patriotismo', 'chilpancingo']*

*['chilpancingo', 'centromedico']*

*['centromedico', 'lazarocardenas']*

*['lazarocardenas', 'chabacano']*

*['chabacano', 'jamaica']*

*['jamaica', 'mixiuhca']*

*['mixiuhca', 'velodromo']*

*['velodromo', 'ciudaddeportiva']*

*['ciudaddeportiva', 'puebla']*

*['puebla', 'pantitlan']*

*['buenavista', 'guerrero']*

*['guerrero', 'garibaldilagunilla']*

*['garibaldilagunilla', 'lagunilla']*

*['lagunilla', 'tepito']*

*['tepito', 'morelos']*

*['morelos', 'sanlazaro']*

*['sanlazaro', 'floresmagon']*

*['floresmagon', 'romerorubio']*

*['romerorubio', 'oceania']*

*['oceania', 'deportivooceania']*

*['deportivooceania', 'bosquedearagon']*

*['bosquedearagon', 'villadearagon']*

*['villadearagon', 'nezahualcoyotl']*

*['nezahualcoyotl', 'impulsora']*

*['impulsora', 'riodelosremedios']*

*['riodelosremedios', 'muzquiz']*

*['muzquiz', 'ecatepec']*

*['ecatepec', 'olimpica']*

*['olimpica', 'plazadearagon']*

*['plazadearagon', 'ciudadazteca']*

*['tlahuac', 'tlatenco']*

*['tlatenco', 'zapotitlan']*

*['zapotitlan', 'nopalera']*

*['nopalera', 'olivos']*

*['olivos', 'tezonco']*

*['tezonco', 'perifericoote']*

*['perifericoote', 'calle11']*

*['calle11', 'tomatlan']*

*['tomatlan', 'esimeculhuacan']*

*['esimeculhuacan', 'puebloculhuacan']*

*['puebloculhuacan', 'atlalilco']*

*['atlalilco', 'mexicaltzingo']*

*['mexicaltzingo', 'ermita']*

*['ermita', 'ejecentral']*

*['ejecentral', 'parquedelosvenados']*

*['parquedelosvenados', 'zapata']*

*['zapata', '20denoviembre']*

*['20denoviembre', 'insurgentessur']*

*['insurgentessur', 'mixcoac']*

2.- Utiliza y/o adapta el código que implementaste en la práctica 6 para realizar la exploración de un grafo mediante el recorrido primero en anchura. Tu función se debe llamar *BusquedaPrimeroAnchura(G,s)* en donde los únicos parámetros son el grafo *G* y el vértice inicial *s* a partir del cual inicia el recorrido. El resultado debe mostrar el recorrido en formato de lista. Por ejemplo, si el vértice inicial es ‘balderas’, entonces el recorrido sería: [‘balderas’, ‘cuauhtemoc’, ‘juarez’, ‘ninosheores’, ‘saltodelagua’, ‘insurgentes’, … ]. Para nodos al mismo nivel de profundidad a partir del vértice inicial, dar prioridad de acuerdo a su orden alfabético, por ejemplo, ‘balderas’ tiene cuatro nodos adyacentes, y el recorrido da prioridad al primero de ellos en orden alfabético (‘cuauhtemoc’).

Muestra tu código *(Valor 2 acierto).*

*# -\*- coding: utf-8 -\*-*

*def main():*

*def createArray():*

*#Lee el archivo lineas.txt y lo pasa a un arreglo*

*lineasMetro = list()*

*with open('lineas.txt') as f:*

*for line in f:*

*line = line.splitlines()*

*if(line != ['']): #ignora los espacios en blanco*

*lineasMetro.append(line[0].split(',')) #crea la lista de listas*

*return lineasMetro*

*def vecinos(Grafo,valor):*

*lista\_vecino=[]*

*for i in range (0,len(Grafo)):*

*for j in range(0,1):*

*if (valor==Grafo[i][0]):*

*lista\_vecino.append(Grafo[i][1])*

*for i in range (len(Grafo)):*

*Grafo[i].reverse()*

*if (valor==Grafo[i][0]):*

*lista\_vecino.append(Grafo[i][1])*

*return lista\_vecino*

*def BPA(Grafo,valor):#recibe al grafo y al valor*

*cola=[valor]*

*print(cola)*

*lista\_BFS=[]*

*while len(cola) !=0:*

*valor=cola.pop(0)*

*lista\_BFS.append(valor)*

*lista\_vecinos=vecinos(Grafo,valor)*

*lista\_vecinos.sort()*

*lista\_vecinos.reverse()*

*for x in range(len(lista\_vecinos)-1,-1,-1):*

*if lista\_vecinos[x] not in lista\_BFS:*

*cola.append(lista\_vecinos[x])*

*cola=Scan(cola)*

*return lista\_BFS*

*def Scan(cola):#revisa que no haya duplicados*

*aux=[]*

*for item in cola:*

*if item not in aux:*

*aux.append(item)*

*return aux*

*lineasMETRO=createArray()*

*print(BPA(lineasMETRO,'balderas'))*

*main()*

Prueba tu código utilizando como vértice inicial ‘balderas’. *(Valor 3 aciertos)*

['balderas', 'cuauhtemoc', 'juarez', 'ninosheroes', 'saltodelagua', 'insurgentes', 'hidalgo', 'hospitalgeneral', 'doctores', 'isabellacatolica', 'sanjuandeletran', 'sevilla', 'bellasartes', 'guerrero', 'revolucion', 'centromedico', 'obrera', 'pinosuarez', 'chapultepec', 'alllende', 'garibaldilagunilla', 'buenavista', 'tlatelolco', 'sancosme', 'chilpancingo', 'etiopiaplazadelatransparencia', 'lazarocardenas', 'chabacano', 'merced', 'sanantonioabad', 'zocalo', 'juanacatlan', 'lagunilla', 'laraza', 'normal', 'patriotismo', 'eugenia', 'jamaica', 'laviga', 'viaducto', 'candelaria', 'allende', 'tacubaya', 'tepito', 'autobusesdelnorte', 'misterios', 'potrero', 'colegiomilitar', 'divisiondelnorte', 'frayservando', 'mixiuhca', 'santaanita', 'xola', 'morelos', 'sanlazaro', 'constituyentes', 'observatorio', 'snpedrodelospinos', 'institutodelpetroleo', 'vallegomez', 'deportivo18demarzo', 'popotla', 'zapata', 'velodromo', 'coyuya', 'villadecortes', 'canaldelnorte', 'floresmagon', 'moctezuma', 'auditorio', 'snantonio', 'lindavista', 'politecnico', 'vallejo', 'consulado', 'indiosverdes', 'lavilla-basilica', 'cuitlahuac', '20denoviembre', 'coyoacan', 'parquedelosvenados', 'ciudaddeportiva', 'iztacalco', 'nativitas', 'romerorubio', 'balbuena', 'polanco', 'mixcoac', 'norte45', 'bondojito', 'eduardomolina', 'martincarrera', 'tacuba', 'insurgentessur', 'viverosderechoshumanos', 'ejecentral', 'puebla', 'apatlaco', 'oceania', 'boulevardpuertoaereo', 'sanjoaquin', 'barrancadelmuerto', 'ferreria', 'talisman', 'aragon', 'panteones', 'refineria', 'miguelangeldequevedo', 'ermita', 'pantitlan', 'aculco', 'deportivooceania', 'terminalaerea', 'gomezfarias', 'azcapotzalco', 'cuatrocaminos', 'camarones', 'copilco', 'generalanaya', 'mexicaltzingo', 'portales', 'hangares', 'zaragoza', 'escuadron201', 'bosquedearagon', 'tezozomoc', 'aquilesserdan', 'universidad', 'tasquena', 'atlalilco', 'nativias', 'villadearagon', 'elrosario', 'iztapalapa', 'puebloculhuacan', 'nezahualcoyotl', 'cdelaestrella', 'esimeculhuacan', 'impulsora', 'uam1', 'tomatlan', 'riodelosremedios', 'constde1917', 'calle11', 'muzquiz', 'perifericoote', 'ecatepec', 'tezonco', 'olimpica', 'olivos', 'plazadearagon', 'nopalera', 'ciudadazteca', 'zapotitlan', 'tlatenco', 'tlahuac']

Prueba tu código utilizando como vértice inicial ‘jamaica’. *(Valor 3 aciertos)*

*['jamaica', 'chabacano', 'frayservando', 'mixiuhca', 'santaanita', 'laviga', 'lazarocardenas', 'obrera', 'sanantonioabad', 'viaducto', 'candelaria', 'velodromo', 'coyuya', 'centromedico', 'doctores', 'pinosuarez', 'xola', 'merced', 'morelos', 'sanlazaro', 'ciudaddeportiva', 'iztacalco', 'chilpancingo', 'etiopiaplazadelatransparencia', 'hospitalgeneral', 'saltodelagua', 'isabellacatolica', 'zocalo', 'villadecortes', 'canaldelnorte', 'tepito', 'floresmagon', 'moctezuma', 'puebla', 'apatlaco', 'patriotismo', 'eugenia', 'ninosheroes', 'balderas', 'sanjuandeletran', 'allende', 'nativitas', 'consulado', 'lagunilla', 'romerorubio', 'balbuena', 'pantitlan', 'aculco', 'tacubaya', 'divisiondelnorte', 'cuauhtemoc', 'juarez', 'bellasartes', 'bondojito', 'eduardomolina', 'vallegomez', 'garibaldilagunilla', 'oceania', 'boulevardpuertoaereo', 'hangares', 'zaragoza', 'escuadron201', 'constituyentes', 'juanacatlan', 'observatorio', 'snpedrodelospinos', 'zapata', 'insurgentes', 'hidalgo', 'alllende', 'talisman', 'aragon', 'misterios', 'guerrero', 'deportivooceania', 'terminalaerea', 'gomezfarias', 'atlalilco', 'auditorio', 'chapultepec', 'snantonio', '20denoviembre', 'coyoacan', 'parquedelosvenados', 'sevilla', 'revolucion', 'martincarrera', 'laraza', 'buenavista', 'tlatelolco', 'bosquedearagon', 'iztapalapa', 'mexicaltzingo', 'puebloculhuacan', 'polanco', 'mixcoac', 'insurgentessur', 'viverosderechoshumanos', 'ejecentral', 'sancosme', 'lavilla-basilica', 'autobusesdelnorte', 'potrero', 'villadearagon', 'cdelaestrella', 'ermita', 'esimeculhuacan', 'sanjoaquin', 'barrancadelmuerto', 'miguelangeldequevedo', 'normal', 'deportivo18demarzo', 'institutodelpetroleo', 'nezahualcoyotl', 'uam1', 'generalanaya', 'portales', 'tomatlan', 'tacuba', 'copilco', 'colegiomilitar', 'indiosverdes', 'lindavista', 'politecnico', 'vallejo', 'impulsora', 'constde1917', 'tasquena', 'nativias', 'calle11', 'cuitlahuac', 'panteones', 'refineria', 'universidad', 'popotla', 'norte45', 'riodelosremedios', 'perifericoote', 'cuatrocaminos', 'camarones', 'ferreria', 'muzquiz', 'tezonco', 'aquilesserdan', 'azcapotzalco', 'ecatepec', 'olivos', 'elrosario', 'tezozomoc', 'olimpica', 'nopalera', 'plazadearagon', 'zapotitlan', 'ciudadazteca', 'tlatenco', 'tlahuac']*

3.- ¿Utilizaste programación orientada a objetos en Python para tu implementación del algoritmo de expansión de grafos?

*NO*

Si tu respuesta es SÍ contesta: ¿Se pudo haber implementado este algoritmo sin utilizar las herramientas de programación orientada a objetos que provee Python? ¿Qué ventajas tiene utilizar el paradigma de programación orientada a objetos en la implementación de este programa?

Si tu respuesta es NO contesta: ¿Qué ventajas consideras que tiene tu implementación con respecto a una que utiliza programación orientada a objetos?

Argumenta claramente tu respuesta (Valor 1 acierto)

*Decidimos hacer este algoritmo sin usar el paradigma orientado a objetos, ya que nos resulta más fácil manejar arreglos de parejas ordenadas de datos, en el que cada elemento de estas, representa un nodo del grafo, y la unión entre éstas, los vértices, relacionando así los nodos, que qrepresentarlos como objetos nodo, a los cuáles se les tendría que agregar atributos para relacionarlos entre ellos.*

*Utilizando ciclos for para iterar sobre las listas, pudimos implementar rápidamente y de manera eficiente el algoritmo de Busqueda Primero en Anchura.*

*La principal ventaja por la que decidimos hacer de esta manera nuestro algoritmo, es la velocidad de implementación y la facilidad para debuggar y corregir errores.*

**SECCIÓN 2.- Implementación de un algoritmo de ordenamiento (*Valor 10 aciertos)***

1.- Utiliza y/o adapta el código de *Bubble Sort* en su versión optimizada con bandera (de la Práctica 1) para ordenar en forma creciente la lista que se generó al realizar el recorrido primero en anchura a partir de la estación ‘balderas’. Incluye un contador del número total de pasadas realizadas y el número de comparaciones. Además, en un archivo llamado *lista.txt* almacena la lista de estaciones ordenadas.

Muestra el código de la función *Bubble Sort (Valor 2 puntos)*

*#BubbleSort Optimizado*

*def bubbleSortOptimized(A):*

*n = len(A)*

*i = 1*

*comparaciones = 0*

*flag = True*

*print("Numero de elementos de la lista: ", n)*

*while i < n-1 and flag:*

*flag = False*

*print "Pasada: " + str(i)*

*for j in range(0, n-i):*

*print(A)*

*comparaciones += 1*

*if A[j] > A[j+1]:*

*temp = A[j]*

*A[j] = A[j+1]*

*A[j+1] = temp*

*flag = True*

*i+=1*

*print("comparaciones: ", comparaciones)*

*print("iteraciones: ", i-1)*

*return A*

*def save(A):*

*archivo = open("lista.txt", "w")*

*for linea in A:*

*archivo.write(linea + "\n")*

*archivo.close()*

*def actividad2():*

*estaciones = #Lee la lista de estaciones*

*listaOrdenada = bubbleSortOptimized(estaciones)*

*save(listaOrdenada)*

*actividad2()*

Muestra aquí el resultado de ejecutar tu programa con la lista del recorrido del grafo utilizando a ‘balderas’ como nodo inicial. Se debe desplegar la lista de estaciones en forma ordenada (Valor 3 puntos), el número de pasadas (Valor 1 punto) y el número de comparaciones (Valor 1 punto).

*['20denoviembre', 'aculco', 'allende', 'alllende', 'apatlaco', 'aquilesserdan', 'aragon', 'atlalilco', 'auditorio', 'autobusesdelnorte', 'azcapotzalco', 'balbuena', 'balderas', 'barrancadelmuerto', 'bellasartes', 'bondojito', 'bosquedearagon', 'boulevardpuertoaereo', 'buenavista', 'calle11', 'camarones', 'canaldelnorte', 'candelaria', 'cdelaestrella', 'centromedico', 'chabacano', 'chapultepec', 'chilpancingo', 'ciudadazteca', 'ciudaddeportiva', 'colegiomilitar', 'constde1917', 'constituyentes', 'consulado', 'copilco', 'coyoacan', 'coyuya', 'cuatrocaminos', 'cuauhtemoc', 'cuitlahuac', 'deportivo18demarzo', 'deportivooceania', 'divisiondelnorte', 'doctores', 'ecatepec', 'eduardomolina', 'ejecentral', 'elrosario', 'ermita', 'escuadron201', 'esimeculhuacan', 'etiopiaplazadelatransparencia', 'eugenia', 'ferreria', 'floresmagon', 'frayservando', 'garibaldilagunilla', 'generalanaya', 'gomezfarias', 'guerrero', 'hangares', 'hidalgo', 'hospitalgeneral', 'impulsora', 'indiosverdes', 'institutodelpetroleo', 'insurgentes', 'insurgentessur', 'isabellacatolica', 'iztacalco', 'iztapalapa', 'jamaica', 'juanacatlan', 'juarez', 'lagunilla', 'laraza', 'laviga', 'lavilla-basilica', 'lazarocardenas', 'lindavista', 'martincarrera', 'merced', 'mexicaltzingo', 'miguelangeldequevedo', 'misterios', 'mixcoac', 'mixiuhca', 'moctezuma', 'morelos', 'muzquiz', 'nativias', 'nativitas', 'nezahualcoyotl', 'ninosheroes', 'nopalera', 'normal', 'norte45', 'obrera', 'observatorio', 'oceania', 'olimpica', 'olivos', 'panteones', 'pantitlan', 'parquedelosvenados', 'patriotismo', 'perifericoote', 'pinosuarez', 'plazadearagon', 'polanco', 'politecnico', 'popotla', 'portales', 'potrero', 'puebla', 'puebloculhuacan', 'refineria', 'revolucion', 'riodelosremedios', 'romerorubio', 'saltodelagua', 'sanantonioabad', 'sancosme', 'sanjoaquin', 'sanjuandeletran', 'sanlazaro', 'santaanita', 'sevilla', 'snantonio', 'snpedrodelospinos', 'tacuba', 'tacubaya', 'talisman', 'tasquena', 'tepito', 'terminalaerea', 'tezonco', 'tezozomoc', 'tlahuac', 'tlatelolco', 'tlatenco', 'tomatlan', 'uam1', 'universidad', 'vallegomez', 'vallejo', 'velodromo', 'viaducto', 'villadearagon', 'villadecortes', 'viverosderechoshumanos', 'xola', 'zapata', 'zapotitlan', 'zaragoza', 'zocalo']*

*('comparaciones: ', 11625)*

*('iteraciones: ', 125)*

Muestra aquí el contenido del archivo *lista.txt* (Valor 1 punto).

20denoviembre

aculco

allende

alllende

apatlaco

aquilesserdan

aragon

atlalilco

auditorio

autobusesdelnorte

azcapotzalco

balbuena

balderas

barrancadelmuerto

bellasartes

bondojito

bosquedearagon

boulevardpuertoaereo

buenavista

calle11

camarones

canaldelnorte

candelaria

cdelaestrella

centromedico

chabacano

chapultepec

chilpancingo

ciudadazteca

ciudaddeportiva

colegiomilitar

constde1917

constituyentes

consulado

copilco

coyoacan

coyuya

cuatrocaminos

cuauhtemoc

cuitlahuac

deportivo18demarzo

deportivooceania

divisiondelnorte

doctores

ecatepec

eduardomolina

ejecentral

elrosario

ermita

escuadron201

esimeculhuacan

etiopiaplazadelatransparencia

eugenia

ferreria

floresmagon

frayservando

garibaldilagunilla

generalanaya

gomezfarias

guerrero

hangares

hidalgo

hospitalgeneral

impulsora

indiosverdes

institutodelpetroleo

insurgentes

insurgentessur

isabellacatolica

iztacalco

iztapalapa

jamaica

juanacatlan

juarez

lagunilla

laraza

laviga

lavilla-basilica

lazarocardenas

lindavista

martincarrera

merced

mexicaltzingo

miguelangeldequevedo

misterios

mixcoac

mixiuhca

moctezuma

morelos

muzquiz

nativias

nativitas

nezahualcoyotl

ninosheroes

nopalera

normal

norte45

obrera

observatorio

oceania

olimpica

olivos

panteones

pantitlan

parquedelosvenados

patriotismo

perifericoote

pinosuarez

plazadearagon

polanco

politecnico

popotla

portales

potrero

puebla

puebloculhuacan

refineria

revolucion

riodelosremedios

romerorubio

saltodelagua

sanantonioabad

sancosme

sanjoaquin

sanjuandeletran

sanlazaro

santaanita

sevilla

snantonio

snpedrodelospinos

tacuba

tacubaya

talisman

tasquena

tepito

terminalaerea

tezonco

tezozomoc

tlahuac

tlatelolco

tlatenco

tomatlan

uam1

universidad

vallegomez

vallejo

velodromo

viaducto

villadearagon

villadecortes

viverosderechoshumanos

xola

zapata

zapotitlan

zaragoza

zocalo

2.- Considerando el tamaño del grafo utilizado en el ejemplo responde las siguientes preguntas. ¿Cuáles de los algoritmos de ordenamiento implementados en las prácticas 1, 2 y 3 hubieran sido más convenientes que Bubble Sort Optimizado con bandera? ¿Cuáles hubieran sido menos convenientes? Justifica claramente tu respuesta. (Valor 2 puntos)

*BubbleSort, aun en su versión optimizada, es un algoritmo poco eficiente cuando se manejan grandes cantidades de datos, ya que tiene una complejidad O(n2). MergeSort tampoco hubiera trabajado de manera eficiente con estos datos.*

*Raddix Sort tampoco es buena opción, debido a la naturaleza de los datos, ya que son cadenas de caracteres. Si convirtiéramos los datos a números, estos serían de un orden muy grande y el algoritmo sería poco eficiente.*

*Couning Sort tampoco es un algoritmo conveniente, ya que no conocemos el rango de valores que pueden tomar los datos y son pocas las repeticiones en los datos, por lo que este algoritmo ocuparía demasiada memoria innecesariamente.*

*Heap Sort y QuickSort serían las mejores alternativas para este problema, debido a su complejidad O(nlogn)*

**SECCIÓN 3.- Implementación de un algoritmo de búsqueda (*Valor 10 aciertos)***

1.- Utiliza y/o adapta el código de Búsqueda Binaria (de la Práctica 4) para determinar si un elemento particular forma parte de la lista de estaciones ordenadas. El programa debe indicar en forma explícita si el elemento buscado se encuentra o no en la lista, y en caso de estar mostrar la posición en la que se encuentra. Incluye un contador del número de comparaciones.

Muestra el código de la función Búsqueda Binaria *(Valor 2 puntos)*

*def Read():*

*file\_=open("lista.txt","r")*

*content = file\_.read()*

*content = content.split("\n")*

*file\_.close()*

*return content*

*def BinarySearch(A, item, start, end,i):*

*if start > end:*

*print("Comparaciones:",i)*

*return False*

*center = int((end-start)/2) + start*

*if A[center] == item:*

*print("Comparaciones:",i)*

*return center + 1*

*else:*

*if item > A[center]:*

*aux=BinarySearch(A, item, center+1, end,i+1)*

*else:*

*aux=BinarySearch(A, item, start, center-1,i+1)*

*return aux*

*Lista = Read()*

*i=0*

*index = BinarySearch(Lista,"balderas",0,len(Lista),i)*

*if index != False:*

*print ("Se encontro el elemento en la posicion:" ,index)*

*else:*

*print ("El elemento no se encuentra en la lista")*

Muestra aquí el resultado de ejecutar tu programa al buscar ‘potrero’. Se debe desplegar claramente si el elemento está o no está en la lista (Valor 2 puntos), la posición en la que se encuentra (Valor 1 punto) y el número de comparaciones realizadas (Valor 1 punto).

*('Comparaciones:', 4)*

*('Se encontro en la posicion:', 114)*

Muestra aquí el resultado de ejecutar tu programa al buscar ‘egipto’. Se debe desplegar claramente que el elemento no forma parte de la lista (Valor 2 puntos), así como el número de comparaciones realizadas (Valor 1 punto).

*('Comparaciones:', 7)*

*El elemento no se encuentra en la lista*

2.- Considerando el tamaño del grafo utilizado en el ejemplo, ¿existe una ventaja al utilizar búsqueda binaria en lugar de búsqueda lineal (Práctica 4)? Justifica claramente tu respuesta. (Valor 1 punto)

*Sí, ya que en la búsqueda lineal se tienen que recorrer todos los elementos hasta encontrar el que se está buscando, lo cual no resulta eficiente con la cantidad de datos que tenemos. Por otra parte, el algoritmo de Binary Search, va acotando los intervalos en los que se busca el elemento, dependiendo si este es mayor o menor que la comparación realizada, acercándose así al elemento de la lista en caso de que éste exista.*

**SECCIÓN 4.- Diseño y realización de un algoritmo paralelo (*Valor 10 aciertos)***

1.- Crear un programa en C que lea el contenido del archivo *lista.txt* creado en la sección 2. El objetivo de este programa será obtener el promedio de la longitud de los nombres de las estaciones. Por ejemplo, ‘balderas’ tiene una longitud de 8 letras, mientras que ‘sanlazaro’ una longitud de 9. Obtener la versión en serie del programa, el cual debe mostrar claramente el tiempo que se tardó en realizar el promedio (la medición del tiempo debe comenzar después de leer los datos del archivo y terminar antes del despliegue del resultado final en pantalla).

Muestra el código *(Valor 1 puntos)*

*Coloca aquí el código completo y borra esta línea*

Muestra aquí el resultado de ejecutarlo. Se debe desplegar claramente el promedio. (Valor 1 punto)

*Coloca aquí el resultado completo que obtienes al ejecutar tu programa y borra esta línea*

2.- Paralelizar el código anterior utilizando OpenMP (Prácticas 11, 12 y 13) con cuatro hilos y sin utilizar la clausula reduction.

Muestra el código *(Valor 2 puntos)*

*Coloca aquí el código completo y borra esta línea*

Muestra aquí el resultado de ejecutarlo. Se debe desplegar claramente el promedio (Valor 1 punto), el número de hilos (Valor 1 punto), qué hilo está haciendo cada parte del proceso (Valor 1 punto) y el tiempo total de ejecución de la región paralela (Valor 1 punto).

*Coloca aquí el resultado completo que obtienes al ejecutar tu programa y borra esta línea*

3.- Utiliza un diagrama *fork-join* para explicar cómo funciona el algoritmo en paralelo. (Valor 1 punto)

*Inserta aquí el diagrama y borra esta línea*

4.- Considerando el tamaño del grafo utilizado en el ejemplo, ¿existe una ventaja al paralelizar el programa? Justifica claramente tu respuesta. (Valor 1 punto)

*Escribe aquí tu respuesta y borra esta línea*