UNSA
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Estructuras Discretas II

Docente: Carlo Corrales Delgado

Actividad N° 4

Ejercicios de Lección 4

Escuela:

Ciencia de la computación (Primer año)

Temas:

-Algoritmo de la ruta más corta

Alumno:

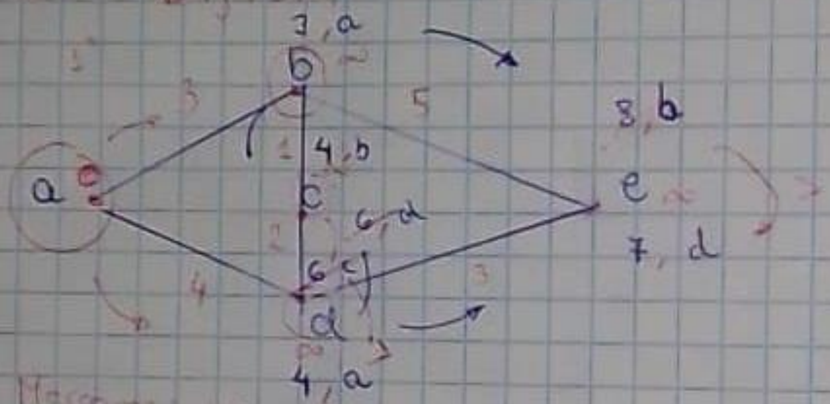
Josue Gabriel Sumare Uscca

Ejercicios de Repaso

1 Describa el algoritmo de trayectoria más corta de Dijkstra

Consta en una serie de pasos en los que etiquetames los vértices de un grafo con ∞ al principio para posteriormente ir cambio por las trayectorias que se van construyendo comparandose entre para dejar aquellas mas pequeñas que se interceptan Siempre se le pone una mara a aquellas etiquetas que terminaron su recorrido.

2 De un ejemplo para mostrar el algoritmo de Dijkstra



Marcamos ya [illegible]
que ya [illegible] desarrollando [illegible]
adyacentes

trayectoria más corta (a, e) = (a, d, e)

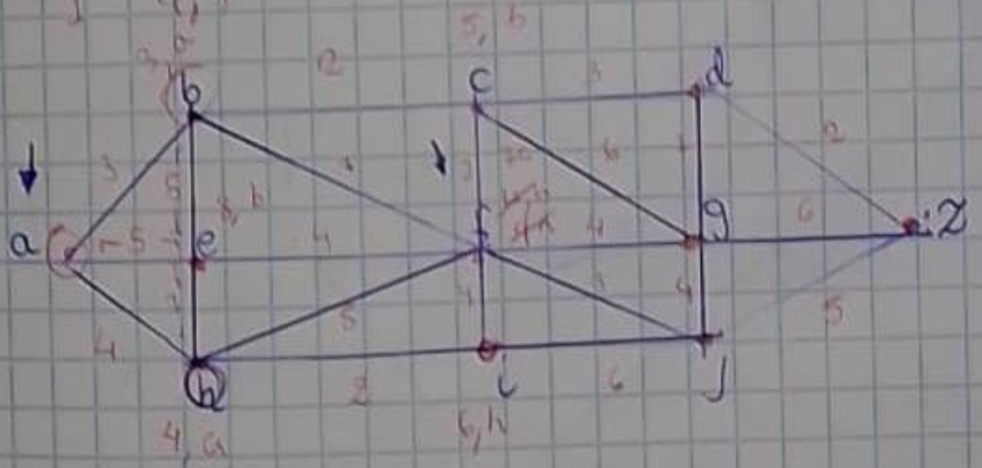
3. Pruebe que el algoritmo de Dijkstra encuentra la una ruta más corta

Con el anterio ejemplo podemos ver que la menor trayectoria efectivamente es la de DijKstra

ruta 1	8
ruta 2	9
ruta 3	12
ruta DijKstra	7

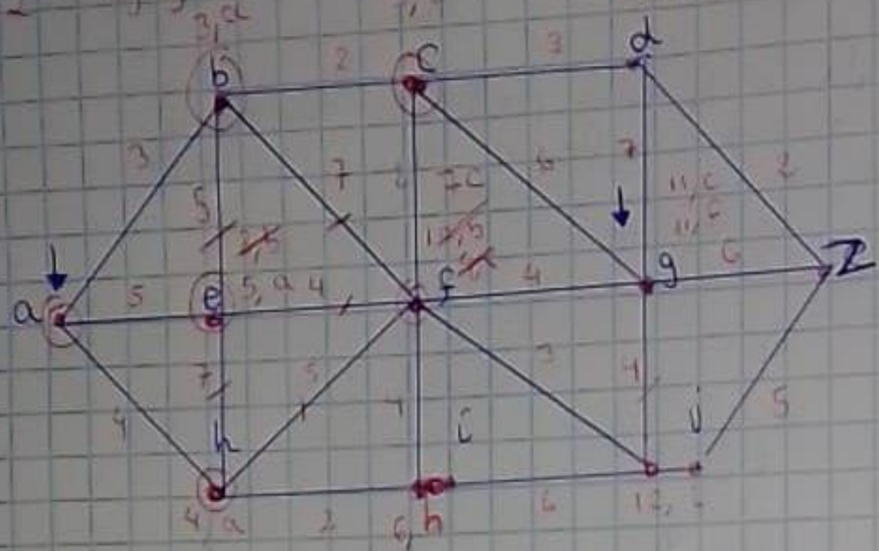
Ejercicio

1



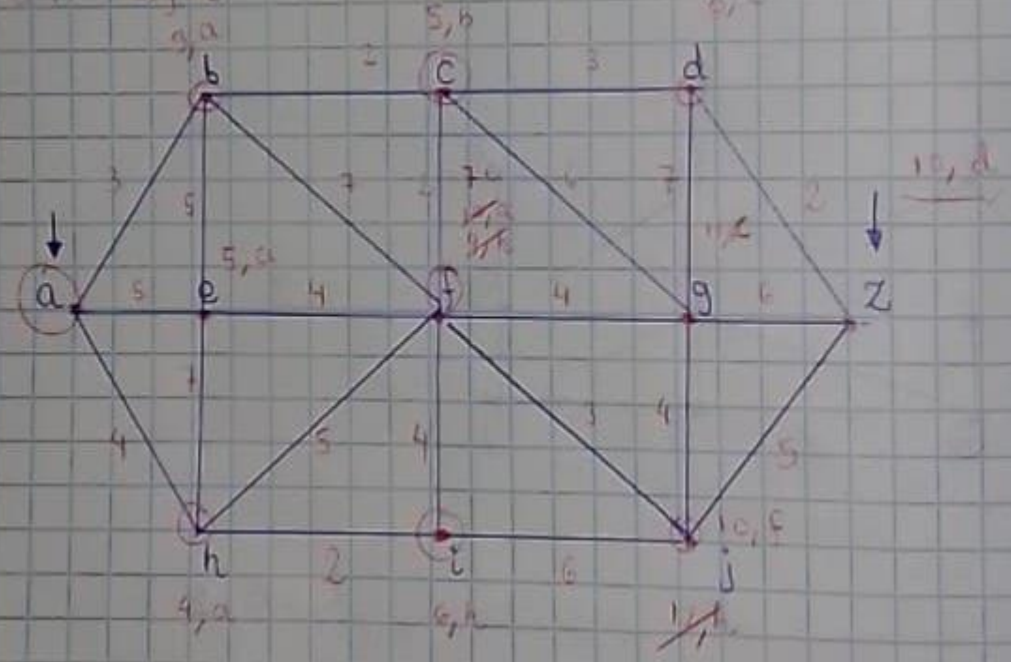
Ruta mas corta (a, f) = (a, 3, b, 2, c, 2, f)

2 a, g

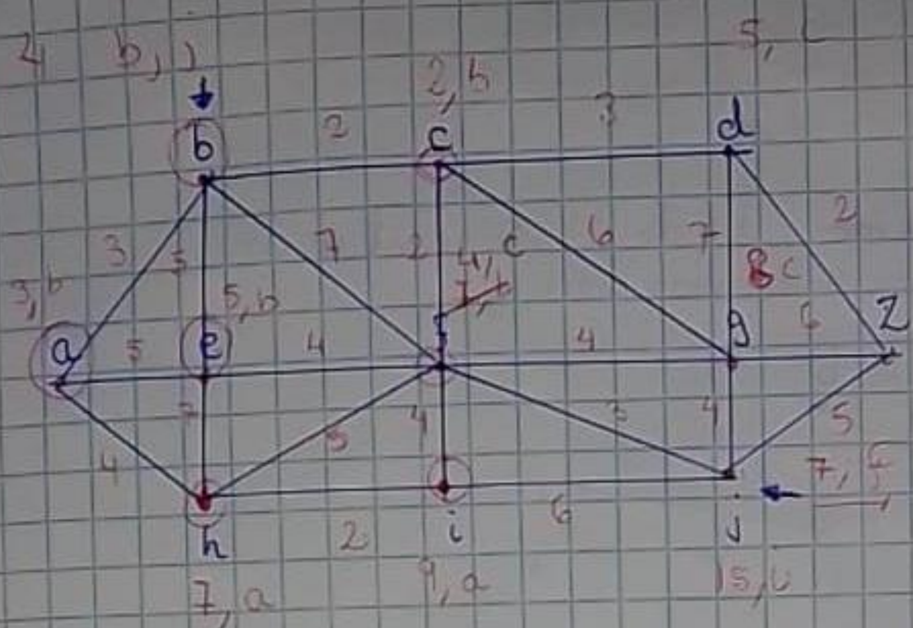


trayectoria mas corta = (a, 3, b, 2, c, 6, g)

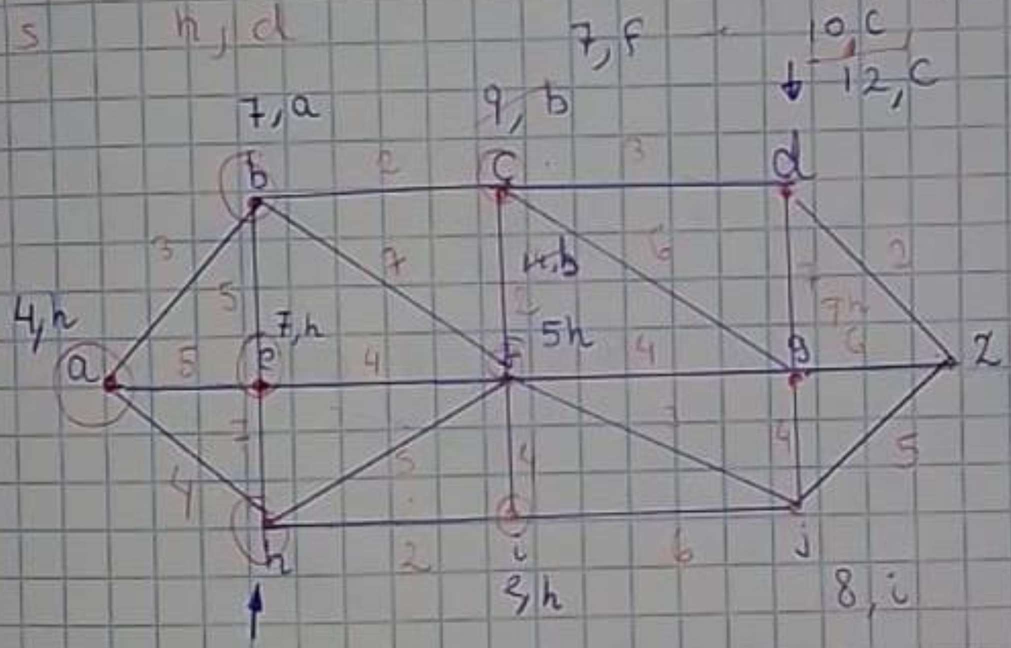
3. a, z



trayectoria mas corta = (a, 3, b, 2, c, 3, d, 2, z)



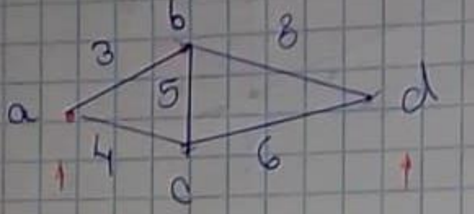
trayectoria mas corta (b, 2, c, 2, f, 3, j)



Ruta mas corta (h, 5, f, 2, c, 3, d)

6. Escribir un algoritmo que encuentre la longitud de ruta mas corta de un vertice a otro

1. Pondriamos todo el conjunto de aristas en un array



aristas = [(a, b), (a, c), (b, c), (b, d), (c, d)]

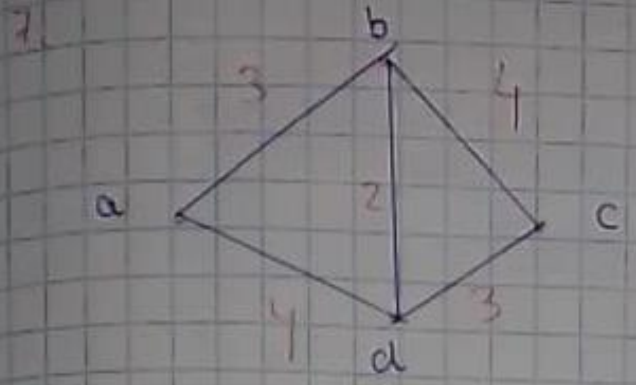
2. Pondriamos los pesos respectivos pesos en otro arreglo colocandolos en los ordenes de la lista anterior

weight = [3, 4, 5, 8, 6]

3. Una vez teniendo este arreglo desarrollamos una funcion que describe el algoritmo anteriormente usado de Dijkstra pasando el parametro de salida (...) y de salida

```
Dijkstra (vertice1, vertice2, aristas, weight):
{ def busqueda (vertice1, Vertice2, aristas
  def busqueda (vertice1, vertice2, weight)
  Algoritmo de Dijkstra
}
```

7.



Dijkstra_1 (vertice, lista de vértices) {

for (ordendelvertice, i ≤ ultimovertice, i++) {

Dijkstra (i (vertice inicial), vertice final, listavertices, listapasos)

}

8. Escribi un algoritmo que encuentra las longitud de las rutas mas [illegible] entre todos los pares [illegible] vertices

1. Utilizamos la anterior función de Dijkstra es solo que sacamos la nº C_2 esto para hallar los pares ordenados vertices de dos vértices y hacemos dos bucles anidados para hallar los pares ordenados

//iteramos una variable en una lista

```
for i in array_vertice:
    for j in arrayvertice
        if i != j
            Dijkstra (i, j, arista weight)
```

9

```
Dijkstra_2 (w, a, z, L) {
        if (Dijkstra (w, a, z, L) != None) {
        return Dijkstra (w, a, z, L)
        else {
            return 0
            }
```

→ Comprobamos si hay una trayectoria de a, z; de no ser así no entraría en la primera estructura condicional y me retornaría 0

10 Al ser una gráfica conexa, sí o sí hay una ruta que une a ambas.

```
dijkstra (w, a, z, L) {
        L(a) = 0
        para todos los vertices x ≠ a
        L(x) = ∞
        T = conjunto de vertices
        while (z ∈ T) {
              seleccionar v ∈ T con L(v) mín
              T = T - {v}
                                        con cero
              para cada x ∈ T adyacente a v
                 L(x) = mín {L(x), L(v) + w(v, x)}
```

Al haber este componente de adyacencia se cumple la condición

11) Opinas sobre el algoritmo 8.4.6 ¿Qué hace? ¿Cómo lo hace?

algor (w, a, z) {

longitud = 0 // inicializa un contador
v = a // una variable que va cambiando
T = conjunto de vértices // arreglo de vértices

while (v != z) { // mientras v ≠ z, o sea mientras no llegue que al final el bucle seguirá

T = T - {v} // sacamos del conjunto el actual elemento
// del conjunto en todos se tomará un vértice con peso mínimo del vértice actual con el mínimo
Seleccionar $x \in T$ con $w(v, x)$ mínima
longitud += $w(v, x)$ // al ser el mínimo
$v = x$ // el valor se le agrega al actualizará al ver contador hasta que llegue
return longitud

}

* Al final del bucle while termina la suma del contador con el camino de vértice a a z más corto

11) Sí, ya que el hallar el mínimo de una ruta los negativos se reconocen y la longitud se suma a este negativo y se lo sigue haciendo mínimo