a)

* Tabla inicial

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | ∞ | 0 | 0 |
| 1 | ∞ | 0 | 0 |
| 2 | ∞ | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | ∞ | 0 | 0 |

* Selecciono Vértice 3
* Actualizo distancia de 0 y 4.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 0 |
| 1 | ∞ | 0 | 0 |
| 2 | ∞ | 0 | 0 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 0 |

* Selecciono Vértice 0
* Actualizo distancia de 2.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | ∞ | 0 | 0 |
| 2 | 2 | P0 | 0 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 0 |

* Selecciono Vértice 2
* No se actualiza distancia de 4.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | ∞ | 0 | 0 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 0 |

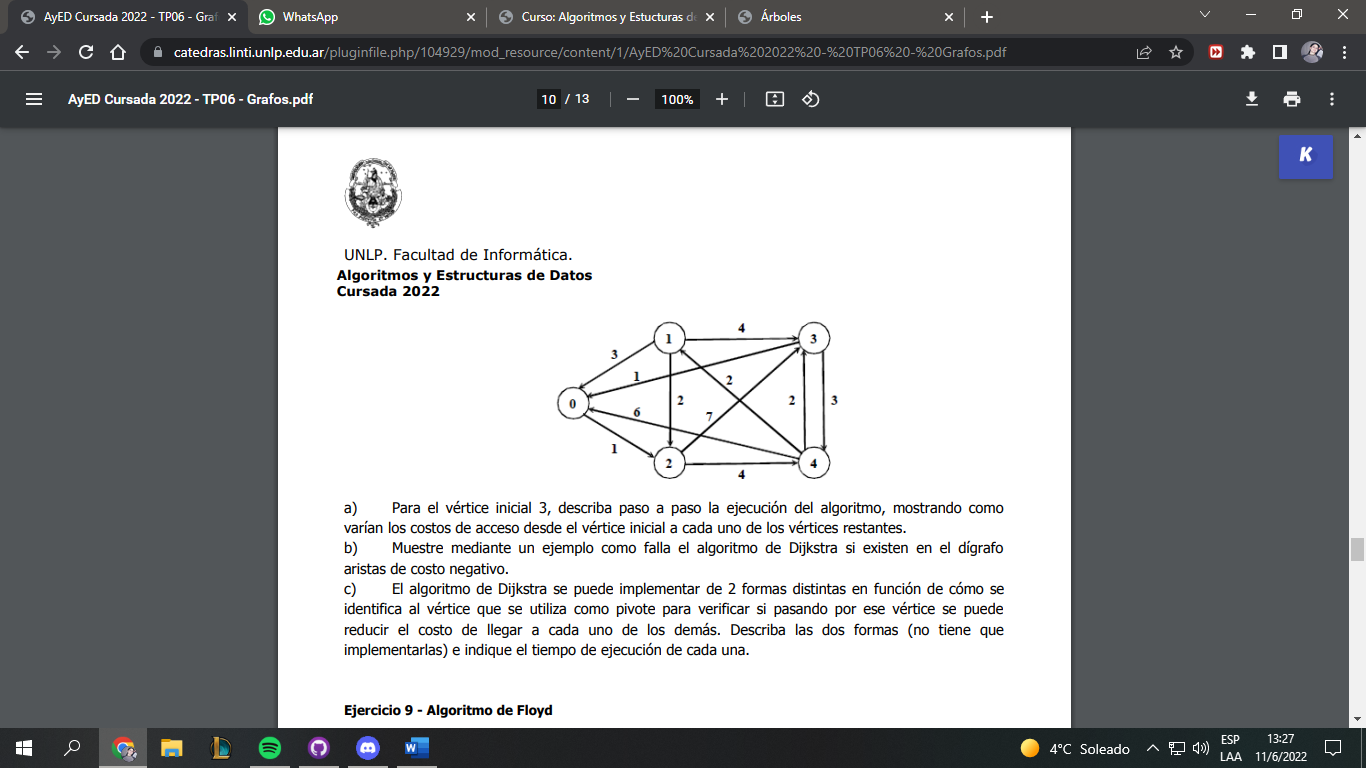
* Selecciono Vértice 4
* Actualizo distancia de 1.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | 5 | P4 | 0 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 1 |

* Selecciono Vértice 1
* No se actualiza distancia de 0 y 2

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | 5 | P4 | 1 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 1 |

b)



* Tabla inicial

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | ∞ | 0 | 0 |
| 1 | ∞ | 0 | 0 |
| 2 | ∞ | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | ∞ | 0 | 0 |

* Selecciono Vértice 3
* Actualizo distancia de 0 y 4.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 0 |
| 1 | ∞ | 0 | 0 |
| 2 | ∞ | 0 | 0 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 0 |

* Selecciono Vértice 0
* Actualizo distancia de 2.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | ∞ | 0 | 0 |
| 2 | 2 | P0 | 0 |
| 3 | 0 | 0 | 1 |
| 4 | 3 | P3 | 0 |

* Selecciono Vértice 2
* Actualizo distancia de 4.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | ∞ | 0 | 0 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | -2 | P2 | 0 |

* Selecciono Vértice 4
* Actualizo distancia de 1.

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | 0 | P4 | 0 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | -2 | P2 | 1 |

* Selecciono Vértice 1

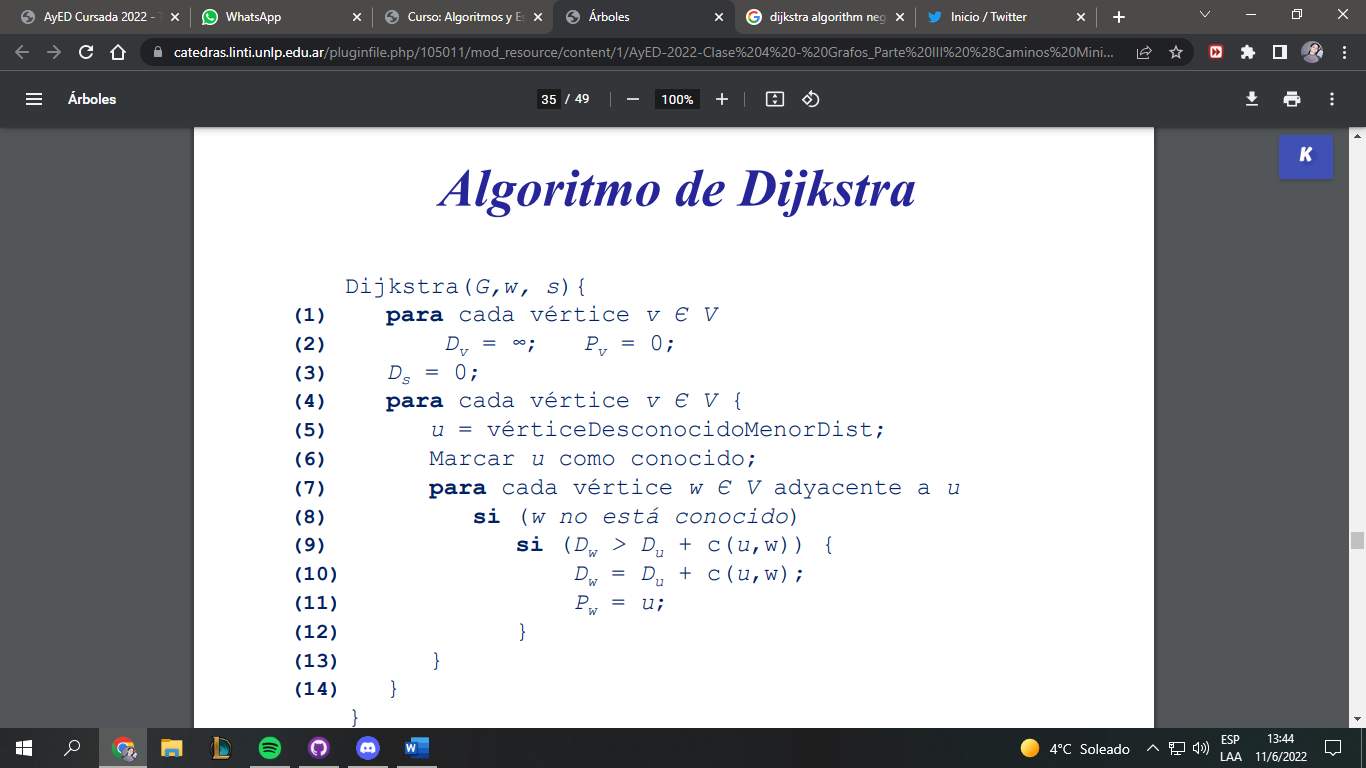
|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | 0 | P4 | 1 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | -2 | P2 | 1 |

ERRORES

|  |  |  |  |
| --- | --- | --- | --- |
| V | DV | PV | Conocido |
| 0 | 1 | P3 | 1 |
| 1 | 0 | P4 | 1 |
| 2 | 2 | P0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | -2 | P2 | 1 |

La distancia mas corta al vértice 0 es pasando por el 4 y después llegando al 0, que daría una distancia de -3. La tabla no se actualiza porque cuando selecciono el vértice 4, el 0 ya esta marcado como visitado.

c)



* El bucle para de la línea (4) se ejecuta para todos los vértices -> |V| iteraciones
* La operación vérticeDesconocidoMenorDist -línea (5)- es O(|V|) y dado que se realiza |V|veces. El costo total de vérticeDesconocidoMenorDist es O(|V|^2 )
* El bucle para de la línea (7) se ejecuta para los vértices adyacentes de cada vértice. El número total de iteraciones será la cantidad de aristas del grafo. |E| iteraciones.
* El costo total del algoritmo es (|V|^2 + |E| ) es O(|V|^2 )

Una optimización es almacenar las distancias en una heap.

* La operación vérticeDesconocidoMenorDist -línea (5)- es O(log|V|) y dado que se realiza |V| veces. El costo total de vérticeDesconocidoMenorDist es O(|V| log |V|)
* El bucle para de la línea (7) que se ejecuta para los vértices adyacentes de cada vértice, también supone modificar la prioridad (distancia) y reorganizar la heap luego de la línea (10). Cada iteración es O(log|V|). Realiza|E| iteraciones, O(|E| log|V|).
* El costo total del algoritmo es (|V| log|V|+ |E| log|V|) es O(|E| log|V|)

La actualización de la heap luego de la línea (10) se puede resolver insertando el vértice w y su nuevo valor Dw cada vez que éste se modifica.

* + El tamaño de la heap puede crecer hasta |E|. Dado que |E| ≤ |V|^2 log |E| ≤ 2 log |V|, el costo total del algoritmo no varía.
  + El costo total del algoritmo es O(|E| log|V|).