INF152 Estructuras Discretas - Tarea 3
Profesores: Claudio Lobos, Jorge Díaz, Sebastián Gallardo, María Paz Vergara, Miguel Guevara.
Ayudantes: Valentina Aróstica, Bryan González, Sofía Mañana, Sofía Riquelme, Carla Herrera, Álvaro Gaete, Constanza Alvarado, Maciel Ripetti, Maureen Gavilán, Ignacio Jorquera, Fernanda Avendaño y Luis González.
Universidad Técnica Federico Santa María
Departamento de Informática – SJ - CC

Tarea 3

1. Reglas generales

- Para resolver cada pregunta, debe hacer uso de los contenidos, algoritmos y métodos aprendidos en el curso. Si su respuesta final es correcta, pero se ha utilizado un método distinto al enseñado en clases no se asignará puntaje. Ídem si entrega resultados sin desarrollo.
- La tarea debe realizarse en grupos de hasta 3 integrantes **DEL MISMO PARALELO**. Esta vez no habrán excepciones.
- Se descontará 5 puntos por warnings en el .tex (Excepto los warnings azules de Overleaf *Underfull \hbox (badness 10000)*, estos no tendrán descuentos).
- Si su proyecto .tex **no** compila tendrá 100 puntos de descuento sin opción a recorrección, inclusive si existe un .pdf con el desarollo de la tarea.
- Tiene hasta las 23:59 hrs del día 19 de junio para entregar esta tarea vía Aula.
- No inserte su desarrollo en fotografías. Debe estar todo desarrollado en LaTex.
- Se restarán 10 puntos de su nota sucesivamente por cada hora de atraso.

1.1. Entrega

- Sólo un integrante del grupo debe realizar la entrega.
- Se debe entregar un solo archivo .zip el cual debe contener:
  1. El o los archivos .tex para compilar su tarea.
  2. Las imagenes adjuntas que permiten compilar el enunciado
  3. Un archivo README.txt que contenga los nombres de los integrantes del grupo, y sus roles.
- El archivo .zip debe tener como nombre: nombre_apellido.zip, donde el nombre es el de el estudiante que hace la entrega.
- No es necesario incluir el .pdf en el .zip

2. Preguntas

Figura 1: El detective Elvis Tek reparando la red neuronal

1. **[50 %]** Luego de pasar la primera prueba de la puerta de seguridad, nuestro detective Elvis Tek, o también conocido por los discretos como Elvis Tikz, recorre Discretilandia hasta llegar a la zona central. Una vez que llega, se percata que la zona de control está destruida y resulta que las conexiones que establecían la red neuronal para manejar a toda Discretilandia, se han caído. Lamentablemente no le queda mucha batería a la fuente de control para restaurar cada actividad, aunque no todo está perdido. Elvis Tikz ha encontrado los planos respecto a las actividades que se deben realizar en la red y la cantidad de energía que consume cada actividad en porcentaje, las que varían entre 0 % y 500 %. Dichas especificaciones aparecen en la tabla 1.

Actividades	A	B	C	D	E	F	G	H	I	J	K
Requisitos	-	A	A	B,K	B,K	B,K	C	F,G	D	E,H	C
Energia	-	-	-	-	-	-	-	-	-	μ	-

Tabla 1: Tabla con actividades y requisitos

Se puede apreciar que el villano que atacó Discretilandia ha borrado el porcentaje de energía que consume cada una de las actividades de la red. Sin embargo, nuestro detective ha encontrado una función Hash que permitiría hallar el consumo de energía de una gran parte de las actividades de la red, **a excepción de la actividad J**.

La función Hash es representada por:

$$\mathcal{H}(x) = \tau \cdot x \tag{1}$$

siendo:

$$\tau = a \; DIV \; b \tag{2}$$

donde $\boldsymbol{a}$ representa la suma de los dígitos verificadores de los roles de todos los integrantes del equipo, $\boldsymbol{b}$ representa la cantidad de integrantes en el equipo y $\boldsymbol{DIV}$ representa al operador división entera. Por otra parte, $\boldsymbol{x}$ representa el índice alfabético de las actividades. Vale decir, el índice alfabético de la actividad A, será 1. El índice alfabético de la actividad B, será 2, y así sucesivamente. Para los integrantes que posean dígito verificador k, utilicen el número 10. Cabe mencionar que si el resultado de $\boldsymbol{\tau}$ les da 0, entonces, en vez de usar el dígito verificador, deben tomar el dígito anterior al dígito verificador de los roles de todos los integrantes, y así sucesivamente hasta que el valor $\boldsymbol{\tau}$ les dé distinto de 0. El detective Elvis Tikz les recuerda que la función Hash retornará la energía que consume la actividad correspondiente, a su vez, les encomienda las siguientes actividades para poder salvar a Discretilandia:

a)[**20 %**] Desencripte la función Hash para hallar la energía que consume cada actividad y, utilizando el package Tikz de LaTeX, construya un grafo con las descripciones de la tabla.
b)[**15 %**] Determine el valor mínimo de μ de manera que la actividad J pertenezca a la ruta crítica de consumo de energía. La ruta crítica debe ser única. Justifique su respuesta y muestre dicha ruta junto con el consumo óptimo total de energía.
c)[**15 %**] Indique, si es que es posible, en qué rango de valores se encontraría μ si para establecer la conexión completa de la red neuronal, se necesita consumir a lo más un 300 % de energía de la fuente de control, y se necesita al menos una holgura de un 20 % para desarrollar la actividad D. En el caso de que no sea posible, fundamente el porqué.

2. [**50 %**] Diseñe un grafo rotulado de al menos 7 nodos, dónde mínimo 5 de los nodos sean de grado mayor a 3 y 2 nodos de grado mayor o igual a 2. Además, debe asignarle peso a los arcos entre los nodos. Una vez listo, realice las siguientes actividades:

   a) [**25 %**] Ejecute el algoritmo Dijkstra y muestre el procedimiento, incluyendo las tablas generadas en las iteraciones y los arcos agregados. Entregue como resultado final la tabla con los caminos más cortos iniciando de un nodo que usted elija. Indique el nodo que se considerará como inicial.

   b) [**25 %**] Bajo el mismo grafo, realice el algoritmo Floyd-Warshall y muestre los pasos a realizar. Entregue como resultado la matriz resultante. ¿Qué pasaría si los arcos pasan de ser no-dirigidos a dirigidos? ¿Cambiarían la forma de la matriz? Fundamente sin realizar el algoritmo nuevamente.

Está demás decir que si se encuentran dos tareas con el mismo grafo y los mismos pesos serán evaluadas con la nota mínima y sin derecho a recorrección.

Nota: todos los grafos de esta tarea deben ser generados utilizando el package Tikz de LaTeX.

3. Desarollo

1. a) Como tengo digito verificador K y hago las tareas solo, el valor de τ es 10 DIV $1 = 10$, lo que deja la funcion hash tal que: $\mathcal{H}(x) = 10 \cdot x$, y esto deja la tabla de actividades como:

Actividades	A	B	C	D	E	F	G	H	I	J	K
Requisitos	-	A	A	B,K	B,K	B,K	C	F,G	D	E,H	C
Energia	10	20	30	40	50	60	70	80	90	μ	110

Tabla 2: Tabla con actividades, requisitos y energías actualizada

Entonces, al crear un grafo con los datos de la tabla anterior, obtenemos:

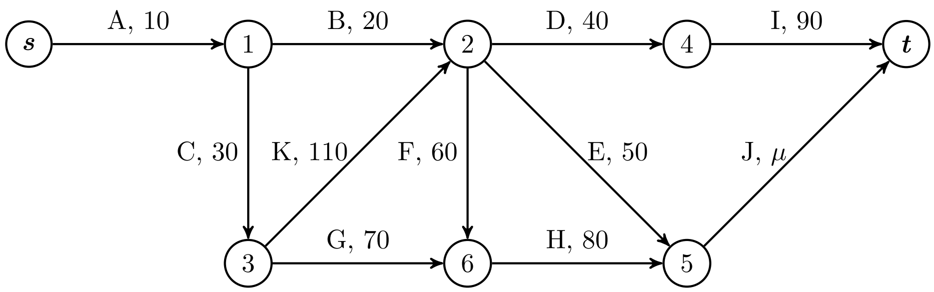

Figura 2: Grafo de energías.

b) Para obtener el valor de mínino de μ para que J pertenezca a la ruta crítica, debemos obtener primero los valores de la mínima energía requerida y la máxima energía que pueden llegar a usar las actividades, recordando que estas tienen valores entre 0 y 500:

$$E(s) = 0,\ por\ definicion$$
$$E(1) = max\{(0 + 10)\} = 10$$
$$E(3) = max\{(10 + 30)\} = 40$$
$$E(2) = max\{(40 + 110), (10 + 20)\} = 150$$
$$E(4) = max\{(150 + 40)\} = 190$$
$$E(6) = max\{(40 + 70), (150 + 60)\} = 210$$
$$E(5) = max\{(210 + 80), (150 + 60)\} = 290$$
$$E(t) = max\{(190 + 90), (290 + \mu)\} = 290 + \mu$$

$$L(t) = E(t) = 290 + \mu,\ por\ definicion.$$
$$L(5) = min\{(290 + \mu - \mu)\} = 290$$
$$L(4) = min\{(290 + \mu - 90)\} = 200 + \mu$$
$$L(6) = min\{(290 - 80)\} = 210$$
$$L(2) = min\{(210-60), (290-50), (200+\mu-40)\} = 150$$
$$L(3) = min\{(150 - 110), (210 - 70)\} = 40$$
$$L(1) = min\{(150 - 20), (40 - 30)\} = 10$$
$$L(s) = min\{(10 - 10)\} = 0$$

Con estos datos podemos armar una tabla que nos ordene los valores recién obtenidos:

x	s	1	2	3	4	5	6	t
$E(x)$	0	10	150	40	190	290	210	290+μ
$L(x)$	0	10	150	40	200+μ	290	210	290+μ

Tabla 3: Tabla con energías mínimas y máximas

Y esta tabla nos permite calcular la holgura, necesaria para obtener la ruta crítica:

Actividad	$L(y) - E(x) - w(x,y)$	$F(x,y)$
A	$10 - 0 - 10$	0
B	$150 - 10 - 20$	120
C	$40 - 10 - 30$	0
D	$(200 + \mu) - 150 - 40$	$10+\mu$
E	$290 - 150 - 50$	90
F	$210 - 150 - 60$	0
G	$210 - 40 - 70$	100
H	$290 - 210 - 80$	0
I	$(290 + \mu) - 190 - 90$	$10+\mu$
J	$(290 + \mu) - 290 - \mu$	0
K	$150 - 40 - 110$	0

Tabla 4: Tabla de holguras

Como podemos ver, la holgura de la actividad J es $(290 + \mu) - 290 - \mu$, por lo tanto, **independiente del valor de μ, J siempre pertenecerá a la ruta crítica**. La ruta crítica está conformada por las actividades con holgura 0, es decir, las actividades {A, C, F, H, J ,K}, y esta ruta crítica tiene un consumo de energía del $290 + \mu\,\%$

c) Si se tiene que consumir como máximo un 300 % y la holgura de D tiene que ser como mínimo un 20 %, entonces podemos encontrar el valor de μ mediante inecuaciones:

$$290 + \mu \leq 300 \qquad\qquad 10 + \mu \geq 20$$
$$\mu \leq 300 - 290 \qquad\qquad \mu \geq 20 - 10$$
$$\mu \leq 10 \qquad\qquad \mu \geq 10$$

Lo que nos deja con la inecuación $10 \leq \mu \leq 10$, la cual se cumple solamente con $\mu = 10$.

2. El grafo que se utilizará será el siguiente:

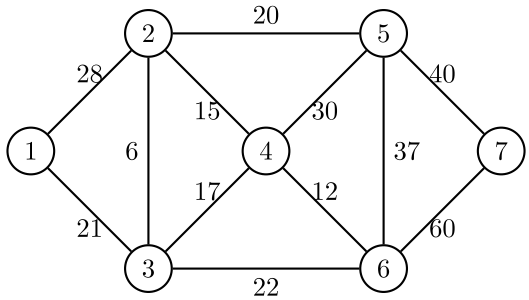

Figura 3: Grafo rotulado.

a) Aplicando Dijkstra para encontrar el camino optimo entre los nodos 1 y 7:

Para aplicar Dijkstra se deben ir registrando y comparando los valores de los arcos adyacentes al nodo que se está evaluando, reemplazando en el caso de que se encuentre un camino más corto. En las tablas de las iteraciones están marcados con negrita el nodo actual, y están tachados los nodos ya visitados, para el nodo actual se elige el que sea menor entre todos los nodos sin visitar de la iteración anterior.

v	$L[v]$	*path*
1	0	-
2	∞	
3	∞	
4	∞	
5	∞	
6	∞	
7	∞	

Tabla 5: Iteración 0

v	$L[v]$	*path*
1	0	-
2	28	1
3	21	1
4	∞	
5	∞	
6	∞	
7	∞	

Tabla 6: Iteración 1

v	$L[v]$	*path*
~~1~~	0	-
2	27	$1 \to 3$
3	21	1
4	38	$1 \to 3$
5	∞	
6	43	$1 \to 3$
7	∞	

Tabla 7: Iteración 2

v	$L[v]$	*path*
~~1~~	0	-
2	27	$1 \to 3$
~~3~~	21	1
4	38	$1 \to 3$
5	47	$1 \to 3 \to 2$
6	43	$1 \to 3$
7	∞	

Tabla 8: Iteración 3

v	$L[v]$	*path*
~~1~~	0	-
~~2~~	27	$1 \to 3$
~~3~~	21	1
4	38	$1 \to 3$
5	47	$1 \to 3 \to 2$
6	43	$1 \to 3$
7	∞	

Tabla 9: Iteración 4

v	$L[v]$	*path*
~~1~~	0	-
~~2~~	27	$1 \to 3$
~~3~~	21	1
~~4~~	38	$1 \to 3$
5	47	$1 \to 3 \to 2$
6	43	$1 \to 3$
7	103	$1 \to 3 \to 6$

Tabla 10: Iteración 5

v	$L[v]$	*path*
~~1~~	0	-
~~2~~	27	1→3
~~3~~	21	1
~~4~~	38	1→3
5	47	1→3→2
~~6~~	43	1→3
7	87	1→3→2→5

Tabla 11: Iteración 6

v	$L[v]$	*path*
~~1~~	0	-
~~2~~	27	1→3
~~3~~	21	1
~~4~~	38	1→3
~~5~~	47	1→3→2
~~6~~	43	1→3
7	87	1→3→2→5

Tabla 12: Iteración 7 y final.

v	$L[v]$	*path*
1	0	-
2	27	1→3
3	21	1
4	38	1→3
5	47	1→3→2
6	43	1→3
7	87	1→3→2→5

Tabla 13: Resultado final después de aplicar Dijkstra.

Entonces, después de ejecutar el algoritmo de Dijkstra en el grafo, tenemos que el camino optimo, o de menor costo, entre los nodos 1 y 7 es $\{1 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 7\}$

b) Aplicando Floyd-Warshall para encontrar el camino mas corto entre cualquier par de nodos

Para aplicar Floyd-Warshall se van comparando los costos de los caminos con el costo de usar un nodo 'auxiliar', es decir, un camino a través de otro nodo, por ejemplo, el camino de 1 a 2 es 28, pero el camino de 1 a 2 pasando por 3 es 27, y así comparando los costos hasta tener una matriz con los menores costos entre todo par de nodos del grafo.

	1	2	3	4	5	6	7
1	0	∞	∞	∞	∞	∞	∞
2	-	0	∞	∞	∞	∞	∞
3	-	-	0	∞	∞	∞	∞
4	-	-	-	0	∞	∞	∞
5	-	-	-	-	0	∞	∞
6	-	-	-	-	-	0	∞
7	-	-	-	-	-	-	0

Tabla 14: Iteración 0.

	1	2	3	4	5	6	7
1	0	28	21	∞	∞	∞	∞
2	-	0	6	15	20	∞	∞
3	-	-	0	17	∞	22	∞
4	-	-	-	0	30	12	∞
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 15: Iteración 1, caminos directos.

	1	2	3	4	5	6	7
1	0	28	21	43	48	∞	∞
2	-	0	6	15	20	∞	∞
3	-	-	0	17	26	22	∞
4	-	-	-	0	30	12	∞
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 16: Iteración 2, pasando por el nodo 2.

	1	2	3	4	5	6	7
1	0	27	21	38	47	43	∞
2	-	0	6	15	20	28	∞
3	-	-	0	17	26	22	∞
4	-	-	-	0	30	12	∞
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 17: Iteración 3, pasando por el nodo 3.

	1	2	3	4	5	6	7
1	0	27	21	38	47	43	∞
2	-	0	6	15	20	27	∞
3	-	-	0	17	26	22	∞
4	-	-	-	0	30	12	∞
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 18: Iteración 4, pasando por el nodo 4.

	1	2	3	4	5	6	7
1	0	27	21	38	47	43	87
2	-	0	6	15	20	27	60
3	-	-	0	17	26	22	66
4	-	-	-	0	30	12	70
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 19: Iteración 5, pasando por el nodo 5.

	1	2	3	4	5	6	7
1	0	27	21	38	47	43	87
2	-	0	6	15	20	27	60
3	-	-	0	17	26	22	66
4	-	-	-	0	30	12	70
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 20: Iteración 6, pasando por el nodo 6.

	1	2	3	4	5	6	7
1	0	27	21	38	47	43	87
2	-	0	6	15	20	27	60
3	-	-	0	17	26	22	66
4	-	-	-	0	30	12	70
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 21: Iteración 7, pasando por el nodo 7 y final.

	1	2	3	4	5	6	7
1	0	27	21	38	47	43	87
2	-	0	6	15	20	27	60
3	-	-	0	17	26	22	66
4	-	-	-	0	30	12	70
5	-	-	-	-	0	37	40
6	-	-	-	-	-	0	60
7	-	-	-	-	-	-	0

Tabla 22: Resultado final después de aplicar Floyd-Warshall.

Y este es el resultado final despues de aplicar el algoritmo de Floyd-Warshall al grafo, notese que es una matriz simétrica y por eso está completa solo la mitad.

Si los arcos del grafo pasaran a ser dirigidos, se deberia aplicar nuevamente el algoritmo puesto que dejaría de ser una matriz simétrica y los caminos de 1 a 3 ya no serian los mismos que de 3 a 1, lo que provocaria el uso de la matriz completa y no solo la mitad.