

Este es el segundo parcial del curso *Árboles y Grafos*, 2025-1. El examen tiene 5 preguntas: otorga un total de 63 puntos y hasta 11 puntos de bono. El examen será evaluado sobre 63 puntos.

El examen es *individual* y debe ser desarrollado a mano. No puede usar el computador ni ningún otro dispositivo electrónico; no puede hablar, chatear o compartir sus soluciones con sus compañeros. Adicionalmente, solo puede sacar durante el parcial las dos hojas de algoritmos escritos a mano de acuerdo con las indicaciones dadas en clase. Además de esto no puede sacar ningún tipo de material adicional en papel y solo puede usar durante el parcial las hojas de cuadernillo entregadas por el profesor. Si necesita una hoja adicional para hacer anotaciones puede solicitarla. Tenga en cuenta los puntos de cada pregunta y planifique adecuadamente su tiempo. El parcial tiene una duración de 140 minutos.

Nombre y código: _____

Pregunta	1	2	3	4	5	Total
Puntos	12	20	15	16	0	63
Puntaje						

Conceptos Teóricos Conectividad y Caminos más Cortos [12 pts.]

1. Responda los siguientes ítems:

- (a) (6 puntos) Sea $G = (V, E)$ un grafo dirigido, $G^T = (V, E)$ el grafo transpuesto de G y $w : E \rightarrow \mathcal{R}$, $w^T = (w^T)^{-1}$ las funciones de pesos asociada a G y G^T respectivamente. Sean $u, v \in V$, si $p = u, x_1, x_2, \dots, x_k, v$ es un camino más corto de u a v en G , ¿es posible que haya un camino más corto de v a u diferente de $q = v, x_k, \dots, x_2, x_1, u$ en G^T ? Explique.
- (b) (6 puntos) Sea $G = (V, E)$ un grafo no dirigido. Considere la arista $(u, v) \in E$ si el nodo u es descubierto antes que el nodo v al hacer un DFS, ¿qué tendría que ocurrir para qué se pueda concluir que la arista (u, v) es un puente?. Explique.

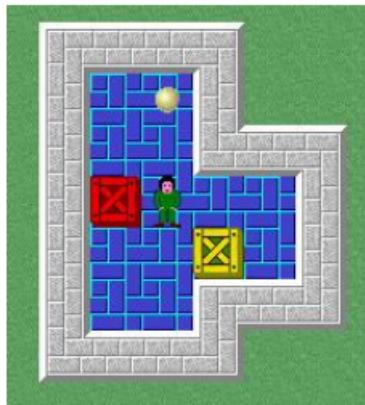
Algoritmos Conectividad y Caminos más Cortos [51 pts.]

2. Considere el siguiente problema:

Sokoban es un popular juego de ordenador creado originalmente en 1982 por Hiroyuki Imabayashi en la empresa japonesa Thinking Rabbit, Inc. Sokoban significa “guardián del almacén” en japonés. La idea es que eres un guardián de almacén que intenta empujar cajas a sus ubicaciones correctas en un almacén. La siguiente imagen muestra el juego.

El objetivo del juego es ayudar al guardián del almacén a empujar las cajas a sus destinos correctos en un almacén abarrotado, para lo que se dispone de un mapa con vista superior. El inconveniente es que nuestro amigo no tiene fuerza suficiente

para levantar una caja sobre otra y tampoco para tirar de ellas. Sólo puede empujar una caja cada vez, dos cajas a la vez ya son demasiado pesadas. Además, un pasillo es tan estrecho como una caja, así que si una caja bloquea un camino, no podrá rodearl simplemente colándose entre la caja y la pared, tendrá que encontrar un camino despejado para llegar al otro lado. Y, por supuesto, nuestro amigo tampoco puede trepar cajas.



Por lo tanto, para ganar este juego, el jugador tiene que guiar al guardián del almacén para que se mueva por el camino correcto a través del laberinto de paredes y cajas y empuje todas las cajas hasta los puntos de destino previstos en el almacén. En Internet se pueden encontrar mapas de Sokoban con distintos niveles de dificultad. Por ejemplo, la figura anterior viene dada por el siguiente mapa:

```
#####
# .#
#   ###
#*@ #
# $ #
#   ###
#####
```

El significado de cada carácter en este mapa es el siguiente: '#' denota una pared, espacio denota una casilla vacía, '.' denota una casilla de meta vacía, '\$' denota una caja en una casilla (una caja amarilla en la imagen), '*' denota una caja en una casilla de meta (una caja roja en la imagen), '@' denota la posición del guardián del almacén en una casilla vacía, '+' denota la posición del guardián del almacén en una casilla de meta.

En estos mapas, estas son las acciones que nuestro pequeño amigo puede realizar:

- Moverse a una casilla vacía adyacente (izquierda, derecha, arriba, abajo, pero no en diagonal). Cada una de estas acciones se denomina *movimiento*.
- Empujar una caja adyacente a una casilla vacía (izquierda, derecha, arriba, abajo, pero no en diagonal). Cada una de estas acciones se denomina *empujón*. Recuerda que para empujar una caja en una dirección, la casilla de destino debe estar vacía.

Dada la configuración inicial de un mapa del juego se debe determinar si es posible ganar y en caso de que sea posible cuál es la menor cantidad de empujones que son necesarios.

- (a) (3 puntos) Explique por qué con los datos suministrados no es posible obtener directamente la respuesta.
- (b) (8 puntos) Indique los elementos que considera se deben tener en cuenta en los estados y para un estado arbitrario cuáles serían las posibles transiciones y las condiciones en los datos del estado para que se den dichas transiciones. Explique.
- (c) (6 puntos) Proponga un ejemplo pequeño de datos de entrada y muestre cómo luciría una parte del grafo de estados para dicho ejemplo, incluyendo el estado inicial y el(s) estado(s) final(es).
- (d) (3 puntos) ¿cuál sería el algoritmo o los algoritmos más apropiados para resolver el problema? Explique.

3. (15 puntos) Considere el siguiente problema:

The International Cargo and Packaging Company Inc. (ICPC Inc.) transports cargo between two cities: Source City and Sink City. ICPC Inc. does not have its own fleet, instead it contracts the service of two train companies, the company *A* and company *B*. Each company has its own network that connects some of the cities at prices that the company decides. For two given cities, it is possible that the route between them is served by both companies, only one company, or none. ICPC Inc. has reached an agreement with both companies that allows it to use their combined services at a discount price, but it has to follow these rules:

1. For a given shipment, ICPC Inc. must specify the percentage of participation of each company given by a for company *A* and $(1 - a)$ for company *B*, for a given real number a ($0 \leq a \leq 1$).
2. When the segment between two cities is served by only one company, ICPC Inc. will pay the price corresponding to that company.
3. When the segment between two cities is served by both companies, the shipment can be shipped using a train from any of the two companies and will pay a fare equal to $a \times C_A + (1 - a) \times C_B$, where C_A and C_B correspond to the fares of company *A* and *B* respectively.
4. A shipment could pass through several intermediate cities. The total cost of the shipment corresponds to the sum of the costs of the individual segments between cities calculated according to rules 2 and 3.

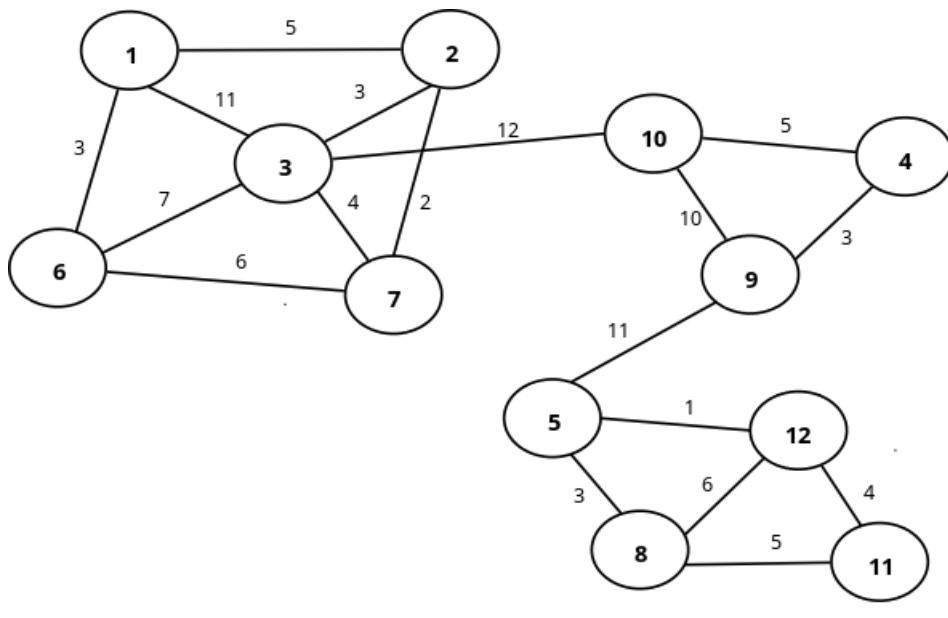
ICPC Inc. needs your help to optimize its costs. Specifically, ICPC Inc. needs to evaluate the cost of different alternatives that combine the participation of the two train companies in different proportions. For this, it is necessary to determine the optimal cost of a trip from city 0 to city $n - 1$, where n is the number of cities and it is assumed cities are indexed by $0, \dots, n - 1$.

Especifique el problema y escriba un algoritmo que resuelva este problema. Indique y explique la complejidad de su solución.

4. (16 puntos) Considere el siguiente problema:

En el AGRAdable y AGRAndado Imperio de Zlatan hay estaciones de tren conectadas por vías en las que los trenes transitan en ambas direcciones. Para cada vía el paso de un tren por ella genera un costo y la suma de los costos de las vías usadas por un tren determina el valor que deben pagar los pasajeros. Las ciudades corresponden a agrupaciones de estaciones en las que para efectos de movilidad se garantiza que desde cualquier estación de tren hay alguna forma de llegar a las demás estaciones. Las ciudades están conectadas entre si por una única vía que funciona en ambas direcciones. Cada estación de tren es capaz de atender una cantidad de pasajeros al día. En cada ciudad hay una estación principal que por lo general es más grande y bonita que todas las demás estaciones. La estación principal de cada ciudad es aquella que puede atender la mayor cantidad de pasajeros en la ciudad.

El siguiente gráfico muestra una parte del imperio:



En el gráfico se observan 3 ciudades. En el recuadro inferior se observa la capacidad de pasajeros para cada estación. La primera ciudad está compuesta por las estaciones 1, 3, 2, 6 y 7. La segunda ciudad está compuesta por las estaciones 10, 4 y 9 y la tercera ciudad está compuesta por las ciudades 5, 12, 8 y 11. La estación principal de la primera ciudad es la estación 7 con 80 pasajeros por día. La estación principal de la segunda ciudad es la estación 4 con 50 pasajeros por día. En tanto que, la estación principal de la tercera ciudad es la estación 8 con una capacidad de 100 pasajeros.

Kenny vive en una de las ciudades y como cuenta con escasos recursos económicos salir de la ciudad es algo que ocurre con poca frecuencia. A Kenny le gustaría conocer las estaciones

principales de cada ciudad y le gustaría poder hacerlo pagando la menor cantidad de dinero posible ya que para lograrlo debe hacer trabajos nada AGRAdables y ahorrar por mucho tiempo. Es por esta razón que a Kenny le gustaría conocer cuál es el costo que debe pagar para llegar a la estación principal de cada ciudad y volver desde la estación que queda más cerca de su casa.

Kenny necesita su ayuda y estará muy AGRAdecido por la colaboración. Especifique el problema y escriba un algoritmo que resuelva este problema. Indique y explique la complejidad de su solución.

5. Elija uno y solo uno de los siguientes ejercicios:

Opción 1: (9+) Escriba un algoritmo que reciba un grafo y use una versión modificada del recorrido primero en profundidad para determinar para cada arista qué rol cumplió durante el recorrido (*tree edge, back edge, forward edge, cross edge*).

Opción 2: (11+) Demuestre la siguiente propiedad:

Sea $G = (V, E)$ un grafo no dirigido y sea T un árbol de DFS de G . Suponga que $r \in V$ es la raíz de T , luego si r tiene al menos dos hijos en T , entonces r es un punto de articulación de G .

Opción 3: (6+) Demuestre la siguiente propiedad:

Sea $G = (V, E)$ un grafo dirigido con una función de peso $w : E \rightarrow \mathbb{R}$ y sea $p = v_0, v_1, \dots, v_k$ un camino más corto de v_0 a v_k entonces todos los subcaminos de p son también caminos más cortos.