

Apellido	Nombre	Legajo	Corrigió
Ejercicio 1:	Ejercicio 2:	Ejercicio 3:	Total:

EJERCICIO 1: Puntaje 5 puntos

Implemente la clase Parcial, y el método:

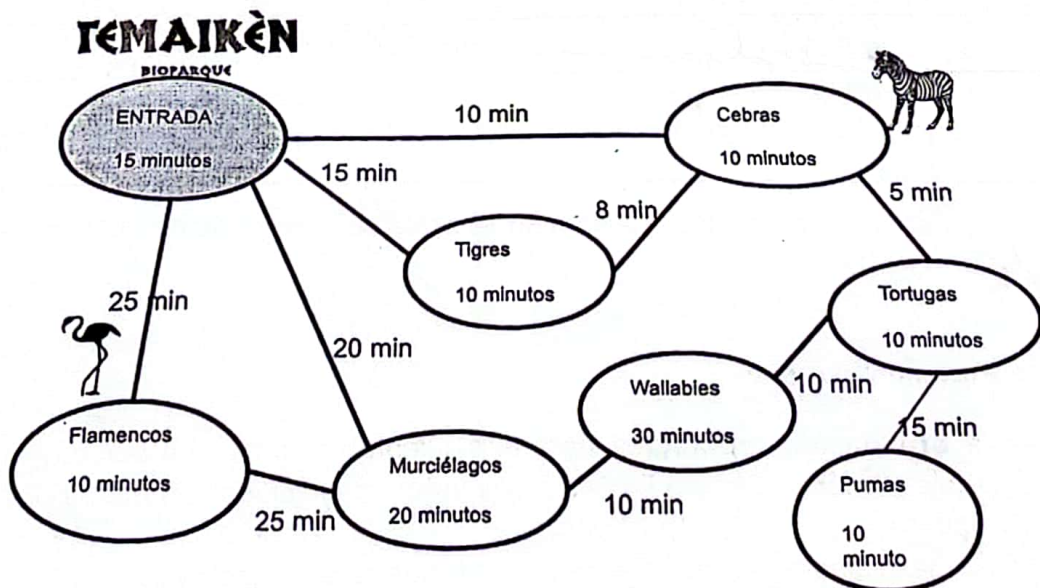
```
public ??? resolver(Graph<???> sitios, int tiempo)
```

Una familia se encuentra planificando su visita al Bioparque Temaikèn en las vacaciones de invierno. Quieren visitar la máxima cantidad de recintos posibles dentro del tiempo que disponen. Se quiere saber la **cantidad máxima de recintos** que se puede visitar en un tiempo determinado partiendo desde la "Entrada".

Cada vértice contiene el nombre del sitio (o recinto de animal) y el tiempo que se tarda en recorrerlo, y en las aristas se cuenta con el tiempo que toma ir de un sitio a otro.

Por ejemplo para este grafo, si el tiempo disponible es **100 minutos**, el método debería devolver 5, ya que con un total de 98 minutos se pueden recorrer los siguientes sitios: 15' (Entrada) + 15' + 10' (Tigres) + 8' + 10' (Cebras) + 5' + 10' (Tortugas) + 15' + 10' (Pumas) = 98'

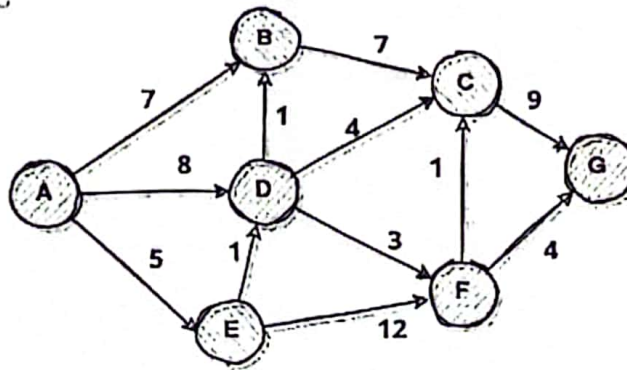
Si el tiempo inicial fuera **30 minutos** solo podría recorrer la "Entrada". Debería devolver 1 vértice.



- Completar en la firma del método los tipos de datos indicados con signo de interrogación.
- La "Entrada" puede no existir.
- **No se puede pasar 2 veces por el mismo lugar.**
- En caso de no existir un recorrido posible, debe devolver 0.
- No se puede agregar variables de clase ni de instancia.

EJERCICIO 2: Puntaje 3 puntos

- a) Muestre paso a paso la ejecución de Dijkstra partiendo del vértice "A". Muestre todos los pasos intermedios, indicando el orden en que se van procesando los vértices.



Orden que toma el Vértice	Vértices	Distancia (A,v)	Vértice Previo	Visitado
1°	A	-	-	1
4°	B	∞ 7	A	0 1
6°	C	∞ 10	D	0 1
3°	D	∞ 6	A	0 1
2°	E	∞ 5	A	0 1
5°	F	∞ 9	D	0 1
7°	G	∞ 13	F	0 1

- b) Marque en el grafo las aristas que forman el árbol con los caminos de costo mínimo que surgen del algoritmo.

EJERCICIO 3: Puntaje 2 puntos

Indicar cuál es la ordenación topológica para el siguiente grafo dirigido acíclico, utilizando la estrategia que trabaja con un recorrido DFS comenzando por el vértice 1. Muestre paso a paso la ejecución del algoritmo.

Nota: considere que las listas de adyacentes están ordenadas en forma ascendente.

