Ejercicios. Programación dinámica

1. Mochila 0-1 con múltiples objetos

Disponemos de infinitos elementos de n tipos distintos $e1,e2,\ldots,en$ con pesos $p1,p2,\ldots,pn$ y beneficios $b1,b2,\ldots,bn$, y tenemos una mochila con capacidad de albergar hasta un máximo de peso M. Queremos encontrar aquellos elementos que tenemos que introducir en la mochila de forma que la suma de los beneficios de los elementos escogidos sea máxima. Los objetos no se pueden fraccionar.

2. Un grupo de amigos van a jugar una partida de trivial. En este caso, han decidido que se van a agrupar en dos equipos, y van a jugar uno contra otro. Entre todos ellos dan por hecho que cuantas más partidas haya jugado cada uno, mejores jugadores son. Por tanto, para que los equipos estén equilibrados, han decidido que la suma de las partidas jugadas por los integrantes de un equipo tiene que ser lo más parecida posible a la suma de las partidas jugadas por el otro equipo.

El grupo de amigos está formado por 12 personas, y cada una ha jugado las siguientes partidas a día de hoy:

- Jugador 1: 3
- Jugador 2: 6
- Jugador 3: 1
- Jugador 4: 4
- Jugador 5: 8
- Jugador 6: 2
- Jugador 7: 5
- Jugador 8: 6
- Jugador 9: 1
- Jugador 10: 4
- Jugador 11: 5
- Jugador 12: 7

Determinar, usando programación dinámica, cómo deben repartirse los jugadores en los dos equipos.

3. En un río hay n embarcaderos. En cada uno de ellos se puede alquilar un bote con el que se puede ir a cualquier otro embarcadero río abajo (es imposible ir río arriba). Existe una tabla de tarifas que indica el coste del viaje del embarcadero i al embarcadero j, para todos los posibles pares (i,j). Puede ocurrir que un viaje de i a j sea más caro que una sucesión de viajes más cortos, en cuyo caso se tomaría un primer bote hasta un embarcadero k y un segundo bote para continuar a partir de k. No hay coste adicional por cambiar de bote.

Dados los siguientes precios, determinar, utilizando programación dinámica, el coste mínimo para cada par de puntos (i,j).

	2	3	4	5	6	7
1	3	4	7	10	15	20
2		4	6	8	13	16
3			1	6	10	12
4				3	7	9
5					2	6
6						5

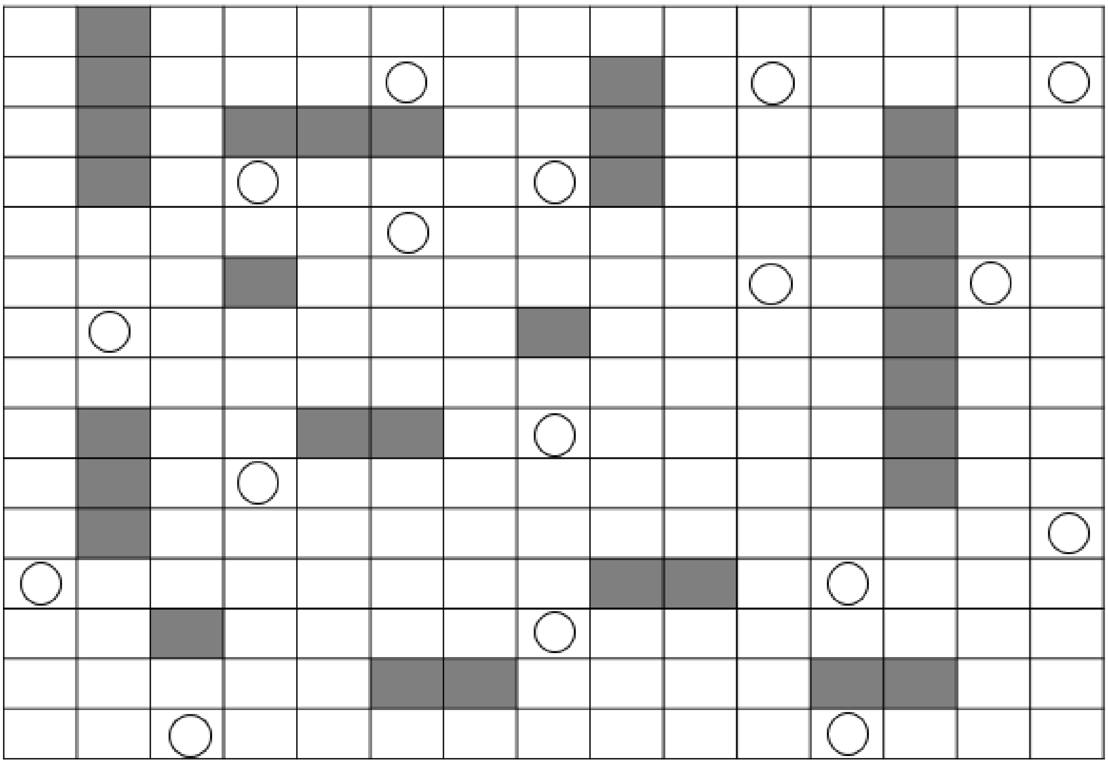
4. Una persona tiene 5000 € para invertir. Como es lógico, desea obtener el máximo rendimiento posible, y para ello ha visitado varios bancos. En cada uno de ellos le han dado un interés diferente dependiendo de la cantidad de dinero que invierta.

El inversor debe decidir cómo repartir el dinero (no todo tiene que estar invertido en el mismo banco) para obtener el mayor beneficio posible.

Los posibles beneficios son:

	1000 €	2000 €	3000 €	4000 €	5000 €
Banco 1	1010	2100	3500	4500	5450
Banco 2	1100	2200	3300	4400	5500
Banco 3	1005	2025	3125	4625	5650
Banco 4	1050	2300	3250	4300	5400
Banco 5	1040	2500	3600	4450	5600

5. En este ejercicio vamos a simular el recorrido de un robot por un laberinto. Este laberinto puede ser visto como una matriz en la que algunas celdas representan una pared. Obviamente, el robot no puede atravesar las celdas que consideramos paredes. Además, en algunas de las celdas que no son pared hay monedas. A continuación se puede ver nuestro laberinto, donde las paredes se muestran como casillas sombreadas y las monedas como círculos.



El recorrido del robot debe comenzar en la esquina superior izquierda de la matriz y debe acabar en la esquina inferior derecha. Además, este robot siempre tiene que hacer el menor número posible de movimientos, es decir, si puede llegar al destino en 28 movimientos, el recorrido elegido no puede tener 29 movimientos o más.

Determina, mediante programación dinámica, el camino que debe seguir el robot para ir desde la casilla de salida (arriba a la izquierda) hasta la casilla de meta (abajo a la derecha) recogiendo el mayor número de monedas posible.