

Resuelve estos problemas **mediante programación dinámica**, indicando:

- 1) La ecuación de recurrencia utilizada, incluyendo casos base.
- 2) Cómo son las tablas necesarias y cómo se rellenan (algoritmo).
- 3) Cómo se reconstruye la solución (algoritmo + describir vector solución).

La ruta de la tapUM. Para la semana BUM 2023 se está preparando esta actividad. Cada estudiante dispone de un cheque-tapa por valor de E euros que puede gastar en los N bares de la guía tapUM. La guía indica para cada bar i un precio fijo p_i para sus tapas, así como un número de tenedores t_i . Para no saturar los bares más valorados, se establece un límite de 3 tapas por bar para cada persona. Debes conseguir el mayor número de tenedores (suma de los de las tapas consumidas) sin superar E . Diseña un algoritmo de PD que indique cuántos tenedores y con cuántas tapas en qué bares se consiguen.

Viajeros a bordo: Tienes una agencia de viajes que organiza excursiones en barco a una famosa isla del Mar Menor. Dispones de dos barcos con capacidad para A y B kg de pasajeros, respectivamente, y de N potenciales pasajeros. Cada pasajero pesa p_i kg y aporta a_i y b_i euros en propinas según se suba al barco A ó B respectivamente. Tu objetivo es organizar un viaje que maximice las propinas sin superar las capacidades de los barcos. Date cuenta que puede que no todos los pasajeros quepan. Diseña un algoritmo de Programación Dinámica que indique el máximo de propinas conseguibles y con qué distribución de pasajeros se consigue esa cifra.

Oktoberfest. A ti y tus amigos os regalan el contenido de un barril de cerveza de L litros. Tenéis N tipos de vasos, cada tipo con una capacidad c_i , y de cada tipo 4 vasos. Diseña un algoritmo de Programación Dinámica que determine qué vasos usar para portar exactamente la cantidad L con el número mínimo de vasos posible.

Ikrea's contest. Para seleccionar a sus trabajadores la cadena Ikrea ha preparado este concurso: Dado un paño de pared de M metros de longitud y N estanterías, cada una con un alto a_i y un ancho b_i , encontrar de cuántas formas se puede llenar la pared con estanterías (se pueden usar a lo alto o a lo ancho). Diseña un algoritmo de Programación Dinámica que devuelva dicho número, así como un ejemplo de forma de colocación.

Puente de bloques. Queremos hacer un puente de bloques de hormigón para cruzar un río de A metros de ancho. Tenemos n bloques de hormigón cuyas medidas, largo x ancho x alto, vienen dadas por una tabla M de dimensiones $3 \times n$. Diseñar un algoritmo de Programación Dinámica que determine qué bloques usar y en qué orientación hacerlo para conseguir un "puente" de longitud A , usando el menor número posible de bloques.