

ANÁLISIS Y DISEÑO DE ALGORITMOS

GRADO EN CIENCIA E INGENIERÍA DE DATOS, CURSO 24/25

PRÁCTICA 4 – PROGRAMACIÓN Y DINÁMICA

A. Contexto

En esta práctica se aplicará la técnica de Programación Dinámica (**PD**) a la resolución de un problema sencillo. El propósito es afianzar la comprensión del funcionamiento de este esquema algorítmico al tener que, no sólo escribir la función recursiva, que es a lo que se presta más atención en las clases de teoría y de problemas, sino también que implementar los pasos de relleno de tabla y reconstrucción de la solución, y aplicarlos a ejemplos de E/S.

B. Enunciado de la actividad

Esta actividad se hará **en grupos de dos alumnos/as**. Cada grupo tiene asignado un problema distinto, según el DNI de los componentes, X e Y (quitando la letra final o la inicial, según el caso). Los problemas asignados dependen del resultado del siguiente cálculo: $R = ((X + Y) \bmod 5) + 1$, donde R indica el número de problema a resolver.

La práctica consistirá en resolver el problema y mostrar algún ejemplo de ejecución utilizando programación dinámica. Es condición necesaria programar el algoritmo propuesto en Python y que funcione correctamente con los casos de prueba que pase el profesor en la revisión de la práctica.

C. Memoria de la actividad

La memoria entregada deberá contener en la portada el profesor de prácticas correspondiente, el nombre de los alumnos, grupo, subgrupo y e-mail. Para el problema asignado al grupo, se deberán incluir los siguientes apartados:

1. (hasta **3** puntos) **Ecuación de recurrencia** necesaria para resolver el problema por programación dinámica, incluyendo casos base e imposibles.
2. (hasta **2** puntos) **Implementación** del algoritmo de construcción (ascendente) de la tabla
3. (hasta **2** puntos) **Implementación** del algoritmo de reconstrucción de la solución a partir de la tabla calculada por el algoritmo del paso 2.
4. (hasta **2** puntos) **Validación** mediante ejemplos sencillos: prueba la ejecución del algoritmo para algunos ejemplos sencillos. Muestra los resultados en la memoria y justifica por qué los resultados obtenidos son óptimos.
5. (hasta **1** puntos) Mini **estudio teórico** del tiempo de ejecución: indica el orden del algoritmo. ¿Hay casos peor/mejor?
6. **Conclusiones y valoraciones personales** de la actividad, incluyendo una estimación en horas del **tiempo** que se ha tardado cada miembro del grupo en completarla.

D. Evaluación de la actividad

La documentación (pdf), el código y cualquier fichero que se considere necesario anexar, se entregarán en un zip a través de la tarea abierta en aulavirtual (entrega un único miembro del grupo). La fecha tope de entrega será **el domingo 15 de diciembre de 2024 a las 23:55**.

El profesor realizará una entrevista individual con cada uno de los alumnos, aunque puede decidir no realizar la entrevista a alguno de ellos si se ha podido realizar un seguimiento suficiente de su trabajo en el laboratorio. La fecha de la entrevista se fijará tras el envío de la documentación.

La puntuación de los distintos apartados son las que se muestran en la sección C de este documento. Es condición necesaria para superar la práctica que el programa implementado haya sido adecuadamente validado.

Respecto a la copia de cualquier ejercicio, de otro grupo o de otra fuente, será de aplicación el Artículo 22 del Reglamento de Evaluación de Estudiantes (REVA)¹ para el alumnado implicado.

E. Problemas

1) La ruta de la tapUPCT. Para la semana de bienvenida 2025 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 tapUPCT. 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.

2) 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 de 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.

3) 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.

¹ El o la estudiante que se valga de conductas fraudulentas, incluida la indebida atribución de identidad o autoría, o esté en posesión de medios o instrumentos que faciliten dichas conductas, obtendrá la calificación de cero en el procedimiento de evaluación y, en su caso, podrá ser objeto de sanción, previa apertura de expediente disciplinario.

4) 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 , el trabajador debe encontrar la forma de llenar la pared con estanterías (se pueden usar a lo alto o a lo ancho) que utiliza un mayor número de estanterías. Diseña un algoritmo de Programación Dinámica que consiga la solución óptima para este encargo.

5) 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.