



UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

CI-1221 Estructuras de Datos y Análisis de Algoritmos
I Ciclo de 2010
Grupo 03

III EXAMEN PARCIAL

Viernes 4 de junio, 5:00 p.m. a 7:30 p.m.

El examen consta de 130 puntos pero no se reconocerán más de 120. Cada pregunta empieza con un indicador de su puntaje y del tema del que trata. Si la pregunta tiene subítemes, el puntaje de cada uno de ellos es indicado al final de la pregunta. Se recomienda echar un vistazo a los temas de las preguntas y a su puntaje antes de empezar a resolver el examen para así distribuir su tiempo y esfuerzo de la mejor manera, de acuerdo a sus destrezas.

I Parte: Solución a problemas utilizando búsqueda exhaustiva, programación dinámica y algoritmos ávidos [110 pts.]

Suponga el siguiente problema. Se necesita crear un algoritmo para una máquina expendedora de bebidas que dé el vuelto utilizando la menor cantidad de monedas. Las denominaciones disponibles son 1, 5, 10 y 25 colones. El algoritmo debe tomar como entrada el monto a retornar m y debe devolver la cantidad de monedas de cada denominación que debe ser devuelta.

1. [35 pts.] Diseñe una solución al problema utilizando la técnica de *búsqueda exhaustiva* por medio de las siguientes subtarefas.
 - (a) Describa la forma de la solución (p. ej., si está dada por un vector, diga qué representa cada una de sus entradas) [5 pts.].
 - (b) Describa el espacio E de soluciones [3 pts.], represéntelo como un producto cartesiano (no olvide definir los conjuntos que forman el producto) [3 pts.], y determine su cardinalidad ($|E|$) [4 pts.].
 - (c) Si es posible, acote el espacio utilizando una restricción del tipo “sujeto a”:
 $E' = \{s \in E : \dots\}$ [5 pts.].
 - (d) Escriba un algoritmo (en pseudo código o C++) que resuelva el problema haciendo una búsqueda exhaustiva con retroceso sobre E o E' [10 pts.] y determine el comportamiento asintótico de la duración del algoritmo (una cota superior es suficiente, además defina claramente la[s] variable[s]) [5 pts.].

2. [40 pts.] Diseñe una solución al problema utilizando la técnica de *programación dinámica* por medio de las siguientes subtareas.
 - (a) Describa cómo puede construirse la solución al problema a partir de soluciones a subproblemas [10 pts.].
 - (b) Escriba un algoritmo que resuelva el problema utilizando programación dinámica. (NOTA: el algoritmo debe devolver el *vector* solución, no basta con devolver la “matriz de escogencias”) [20 pts.].
 - (c) Determine el comportamiento asintótico de la duración del algoritmo [10 pts.].
3. [35 pts.] Determine si es posible crear un algoritmo ávido que devuelva la solución óptima y cuya duración sea $O(n \log n)$. Para ello realice las siguientes subtareas.
 - (a) Diga si es posible crear tal algoritmo y justifique su respuesta [10 pts.].
 - (b) Si es posible crear el algoritmo:
 - i. Escriba el (pseudo) código del mismo y haga notar claramente cualquier asunción acerca de los datos (p. ej., si se requiere que estén ordenados de acuerdo a algún criterio) [10 pts.].
 - ii. Demuestre que el algoritmo es correcto (sugerencia: utilice el método de la *transformación*) [15 pts.].

Si no es posible crear el algoritmo:

 - i. Identifique un argumento que sugiera la posibilidad de crear un algoritmo ávido [10 pts.].
 - ii. Demuestre que tal algoritmo no necesariamente produce soluciones óptimas (sugerencia: encuentre un contraejemplo) [15 pts.].

II Parte: Árboles B [20 pts.]

1. [20 pts.] Suponga que el método B-TREE-SEARCH es implementado utilizando una búsqueda binaria en vez de una búsqueda lineal dentro de cada nodo. Demuestre que este cambio hace que el tiempo de CPU requerido sea $O(\log n)$, independientemente de cómo t (el grado mínimo del árbol) sea seleccionado como función de n (la cantidad de llaves en el árbol).