

Segundo Parcial de Programación 3 (26/11/2012)

Instituto de Computación, Facultad de Ingeniería

- Este parcial dura **4** horas y contiene 2 carillas. El total de puntos es **60**.
- En los enunciados llamamos C^* a la extensión de C al que se agrega el operador de pasaje por referencia &, y las sentencias *new*, *delete* y el uso de *cout* y *cin* y el tipo `bool` predefinido en C++.
- NO se puede utilizar ningún tipo de material de consulta. Salvo que se indique lo contrario podrá usarse todo lo visto en el teórico, práctico y laboratorio sin demostrarlo, indicando claramente lo que se está usando.

Se requiere:

- i. Numerar todas las hojas e incluir en cada una el nombre y la cédula de identidad.
- ii. Utilizar las hojas de un sólo lado y escribir con lápiz.
- iii. Iniciar cada ejercicio en hoja nueva.
- iv. Poner en la carátula la cantidad de hojas entregadas, y un índice indicando en qué hojas respondió cada problema.

Ejercicio 1. (20 puntos)

Una empresa de distribución de mercaderías desea automatizar la asignación de carga de sus vehículos de manera de **utilizar exactamente** su capacidad máxima de carga (por vehículo). La capacidad de cada uno de los vehículos es de C m³. La empresa cuenta con $n+1$ productos, $0 \leq i \leq n$, cuyo volumen se encuentra definido en $T[i]$, siendo $T[i] < C$. La empresa cuenta con un stock $Q[i]$ de cada producto. Para reducir costos de manipulación de stock, se desea que la asignación por vehículo minimice **la cantidad total de unidades de todos los productos** que transporta. En esta instancia interesa **solamente** solucionar el problema de asignar la carga de un **único vehículo**. Tampoco se maneja el concepto de pedido, sólo la carga de un camión lo mejor posible.

Notar que el problema puede no tener solución. Como ejemplos considere los casos $T = [4,5]$, $Q[1,5]$ y $C = 11$ o $C = 12$ para los que no hay solución.

- a) (14 puntos) Formalizar el problema aplicando **Programación Dinámica**, especificando la recurrencia para solucionar el problema de asignar carga a **un** único vehículo mediante la fórmula recursiva $f(p, c)$, donde p indica el uso de los productos desde el 0 hasta el p y c indica la capacidad disponible. Asuma que los volúmenes de los distintos productos se encuentran ordenados de forma creciente en T . En caso de que no haya solución al problema, la recurrencia debe retornar $+\infty$. Tenga en cuenta que no necesariamente todo el stock es asignado. Como justificación de la recurrencia propuesta describa claramente qué se considera en cada ítem. Indique cómo debe invocarse la recurrencia de forma de resolver el problema pedido.
- b) (6 puntos) (sólo se tendrá en cuenta esta parte si realizó la parte (a)) Describa la estructura de datos necesaria para resolver la recurrencia de la parte anterior. Implemente en forma iterativa, en C^* , el algoritmo que la resuelve e indique el orden del tiempo de ejecución para el peor caso del algoritmo implementado, en función de n y C .

Ejercicio 2. (20 puntos)

Parte a) (5 puntos)

- 1) Describa la forma general de la técnica **Greedy** explicando cada paso.
- 2) ¿Qué problema resuelve el algoritmo de **Kruskal**? Describa la estrategia e indique las correspondencias entre el algoritmo y la forma general de la técnica que utiliza.

Parte b) (15 puntos)

- 1) Implemente el algoritmo MergeSort en un lenguaje de alto nivel (C* y/o pseudocódigo como visto en el material teórico es suficiente).
- 2) Defina el concepto de árbol de decisión.
- 3) Considerando la entrada $A = [a, b, c]$, dibuje el árbol de decisión del algoritmo de la parte (b1). Justifique.
- 4) Calcule la cantidad de comparaciones realizadas por el algoritmo MergeSort de (b1) en el peor caso. Suponga n potencia de 2. Justifique los términos de la ecuación de partida en base al algoritmo.
- 5) Defina *estabilidad* de un algoritmo de ordenación, el algoritmo MergeSort de (b1) ¿es estable? Justifique.

Ejercicio 3. (20 puntos)

Un Sistema Operativo es el software que interactúa tanto con el hardware del computador como con las aplicaciones de los usuarios. Su responsabilidad es gestionar los distintos recursos de la computadora.

Un proceso es un programa en ejecución, y durante la misma necesitará de diferentes recursos, que solicitará al Sistema Operativo y éste decidirá cuando otorgárselos. Un recurso es un elemento del computador que los procesos usarán para lograr algún objetivo (por ejemplo: una impresora). Todos los procesos utilizarán todos los recursos en algún orden y durante 1 unidad de tiempo.

Para cada proceso i se cuenta con un vector R_i , que indica el orden en el que solicita los recursos.

- $R_i = \langle R_{i_0}, \dots, R_{i_{n-1}} \rangle$, donde R_{i_j} identifica al recurso que utiliza el proceso en el lugar j .

Se considera despreciable todo tiempo que no sea a la espera de la asignación de un recurso (en particular se despreciará el tiempo de ejecución por el o los procesadores).

Cada recurso puede estar asignado a un único proceso a la vez, esto significa que si otro proceso también necesita utilizar el mismo recurso entonces estará en espera por el recurso.

Se define *tiempo en el sistema* de un proceso, como el tiempo desde el inicio de su ejecución hasta que libera el último recurso luego de utilizarlo (es el tiempo que estuvo utilizando recursos sumado a los tiempos que estuvo a la espera de recursos).

El *tiempo total de ejecución* es la suma de los *tiempos en el sistema* de todos los procesos.

Para cada recurso se desea saber en qué orden lo usarán los procesos, de forma de minimizar el *tiempo total de ejecución*.

Consideraciones:

- No existen dependencias entre los procesos.
 - Existen P procesos identificados entre 0 y $P-1$ y R recursos que se identifican de 0 a $R-1$.
- 1) (12 puntos) Formalizar el problema aplicando *Backtracking*, indicando forma de la solución, restricciones explícitas, restricciones implícitas, función objetivo y predicados de poda en caso que corresponda. En cada ítem explique brevemente en lenguaje natural su significado.
 - 2) (4 puntos) ¿Cuál es la diferencia entre restricciones explícitas y restricciones implícitas? Defina previamente los conceptos involucrados.
 - 3) (4 puntos) Explique la diferencia entre función de poda en la implementación y predicado de poda.