

# Examen de Programación 3 y III (22/07/2011)

## Instituto de Computación, Facultad de Ingeniería, UDELAR

1. Este exámen dura 4 horas y contiene 3 carillas. El total de puntos es 100. Para su aprobación necesita 60 puntos.
2. 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*.
3. NO se puede utilizar ningún tipo de material de consulta.
4. No se contestaran dudas durante la última media hora.

### Se requiere:

- Numerar e incluir el nombre completo y cédula de identidad en la esquina superior derecha de cada hoja.
- Utilizar las hojas de un sólo lado y escribir con lápiz, iniciando cada ejercicio en hoja nueva.
- Completar la carátula con la cantidad de hojas entregadas, y un índice indicando en qué hoja se respondió cada ejercicio y problema.

## Parte Teórica Obligatoria

Esta parte es eliminatoria, vale **40** puntos y usted debe obtener como mínimo el **50% de esta parte (20 puntos)**. En caso de no llegar a dicho mínimo, **NO** se corregirán los problemas. Usted podrá encontrar planteos prácticos. Los mismos deben ser resueltos justificando detalladamente la correspondencia con la base teórica que utilice en su respuesta.

### Ejercicio 1 (10 puntos)

Dadas las siguientes funciones arbitrarias  $f : N \rightarrow R^*$ ,  $g : N \rightarrow R^*$  y partiendo de las definiciones de límite,  $\theta$ ,  $\Omega$  y  $O$ :

Pruebe que si el  $\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = K \in R^+$  entonces  $f(n) \in \Theta(g(n))$  y  $g(n) \in \Theta(f(n))$ .

### Ejercicio 2 (10 puntos)

1) Dadas dos matrices de enteros  $A_{m \times n}$  y  $B_{n \times p}$  representadas en variables que son de tipo arreglos de enteros de dos dimensiones siendo primero la fila y luego la columna. Implementar el procedimiento que calcula el producto de las matrices A y B.

```
// Calcula el producto de la matriz a por la matriz b y pone el resultado en c.  
// La memoria de c fue previamente reservada.  
void producto (int m, int n, int p, int** a, int** b, int** c);
```

- a) Calcular el costo exacto de la implementación de producto de matrices considerando como operación básica la multiplicación.
- b) Si se tiene como dato que además  $\Theta(m) = \Theta(n) = \Theta(p)$ , muestre el orden exacto del costo de su algoritmo en la forma más simple posible. Justifique utilizando propiedades probadas en el curso.

### 2) Método de Strassen:

Para esta parte suponga que las matrices A y B son cuadradas de forma  $A_{n \times n}$  y  $B_{n \times n}$  siendo  $n$  una potencia de dos.

Explique como puede optimizar la multiplicación de matrices utilizando un procedimiento de *Divide and Conquer*. Se deben entender como son los pasos a nivel de operaciones matriciales. *No es necesario que proporcione código.*

### Ejercicio 3 (10 puntos)

1. Describa la forma general de la técnica Greedy en pseudocódigo. Explique cada paso.
2. Explique qué resuelve el algoritmo de Dijkstra. Proporcione un *pseudocódigo*.

### Ejercicio 4 (10 puntos)

1. Defina Árbol de Decisión e indique que representan sus componentes.
2. Implemente un Insertion Sort estable. Justifique porque su implementación es estable.

## Problemas

### Problema 1 (30 puntos)

Una empresa constructora debe llevar a cabo  $O$  obras. Cada obra  $o_i$  tiene  $t_i$  trabajos los cuales serán ejecutados por personal de la empresa. Contractualmente cada obra  $o_i$  debe ser finalizada en  $d_i$  días máximo. En caso que la obra  $o_i$  lleve más días que el máximo establecido en el contrato, se incurre en una penalización  $p_i$  por cada día de atraso. El personal de la empresa se divide en dos grupos,  $A$  de tamaño  $|A|$  y  $B$  de tamaño  $|B|$ . El personal del grupo  $A$  realiza  $A_d$  trabajos diarios a un costo  $A_c$  diario, mientras que el grupo  $B$  realiza  $B_d$  trabajos diarios a un costo  $B_c$  diario. Por política de la empresa toda obra debe contar con al menos  $A_m$  personal del grupo  $A$  y  $B_m$  personal del grupo  $B$ , y no puede tener más de  $A_M$  y  $B_M$  personal del grupo  $A$  y  $B$  respectivamente. La empresa busca asignar el personal de manera de minimizar el costo de ejecutar todas las obras.

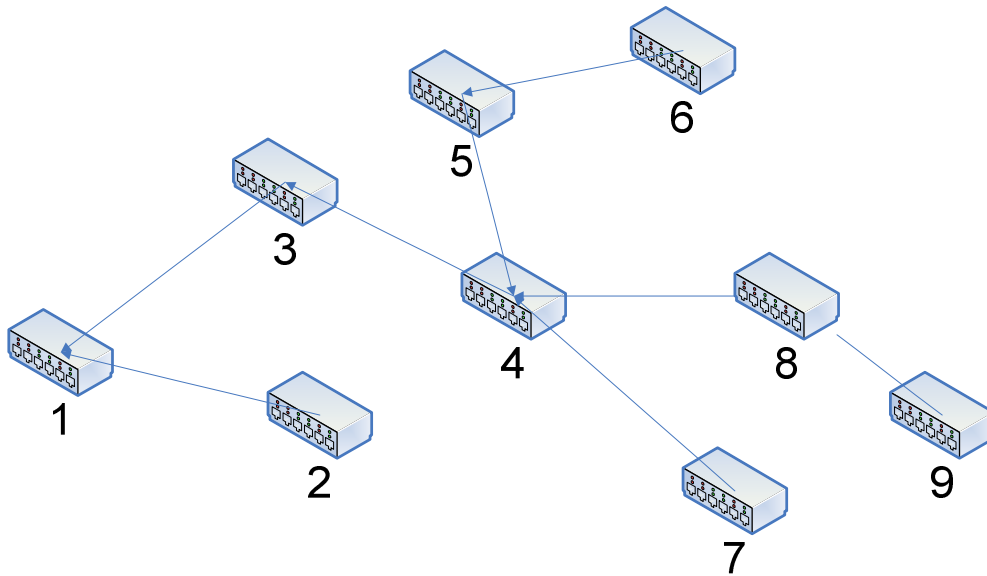
#### Se pide:

- 1) Formalizar el problema para la estrategia de Backtracking indicando: forma de la tupla, restricciones implícitas y explícitas, y función objetivo si corresponde (además explique todos los ítems en lenguaje natural).
- 2) Argumente de forma breve si las siguientes condiciones son o no predicados de poda (considere  $o_i$  como la obra actual, es decir la siguiente a asignar personal):
  - i. El costo acumulado hasta la obra  $o_i$  es mayor que el costo de la mejor solución encontrada hasta el momento.
  - ii. No considerar para la obra  $o_i$  personal de un grupo que fue totalmente asignado.
- 3) Responda las siguientes preguntas;
  - i. Supongamos que el tamaño de los grupos  $A$  y  $B$  es infinito, que no hay cota superior de personal de los grupos para las obras y que  $A_i = kB_i$  con  $k$  entero y mayor a 0. ¿cuál es el mejor grupo de personal a seleccionar y porque? ¿cómo se ve alterada la formalización de la parte a.1?
  - ii. Supongamos que no hay cota inferior de personal de los grupos para las obras, ¿cómo se ve alterada la formalización de la parte a.1?

## Problema 2 (30 puntos)

Una pequeña compañía proveedora de conexión a internet desea asegurar la confiabilidad de conexión a través su red de routers, actualmente biconexa. La compañía cuenta con una alarma que notifica en caso de pérdida de conexión entre al menos un par de nodos a la única cuadrilla de técnicos con la que cuenta la empresa. Dado que muchos de los dispositivos son poco confiables, se desea anticipar dicha alarma, con un proceso que monitoree la red y revele qué dispositivos pasaron a ser críticos (i.e. su caída implicaría la desconexión de al menos un par de nodos), para realizar la reparación correspondiente y volver a garantizar la conectividad biconexa de toda la red cuanto antes. La atención de routers críticos deberá ser priorizada considerando la cantidad de nodos de las subredes que quedarían desconectadas si este cayera, utilizando la siguiente expresión.

$prioridadAtencion(a) = \prod |V_i|$ , siendo  $V_i$  el conjunto de routers en la subred  $i$  (generada por la ausencia de  $a$ ).



e.g. Considerando que la figura representa el estado actual de la red, en el caso que cayera el router 4, quedarían 3 subredes y su prioridad de atención estaría dada por:  $prioridadAtencion(4) = 3 \times 2 \times 3 = 18$ . Análogamente,  $prioridadAtencion(1)=7$ ,  $prioridadAtencion(3)=12$ , etc.

Para ello, se ejecuta periódicamente el proceso *Lista puntosCriticos(Grafo red)* que recibe un *Grafo* que modela el estado actual de la red y retorna el listado de dispositivos críticos ordenados descendientemente por prioridad de atención (a mayor valor, mayor prioridad). No existe prioridad definida entre dispositivos cuya prioridad de atención coincida.

**Se pide:**

4) **Implemente** el proceso *Lista puntosCriticos(Grafo red)*.

Puede asumir la existencia del TAD *Grafo* y *Lista de Enteros*. También puede utilizar la función *insertarDescendente(Lista l, int id, int prioridad)* que inserta el valor *id* en la lista *l* de acuerdo a su *prioridad* para que la misma quede ordenada descendientemente.

La función *prioridadAtencion()* NO es dada. Si desea utilizarla deberá implementarla, pudiéndole agregar todos los parámetros que necesite.

Nota: se recomienda realizar un grafo de cubrimiento mínimo para analizar las subredes.