

# Examen de Programación 3 y III (07/02/2015)

Instituto de Computación, Facultad de Ingeniería, UdelaR

1. Este examen dura 4 horas y contiene 4 carillas. El total de puntos es 100 y se requieren 60 para su aprobación.
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*, el uso de *cout* y *cin* y el tipo *bool*.
3. NO se puede utilizar ningún tipo de material de consulta.
4. No se contestarán dudas durante la última media hora.

## Se requiere:

- Numerar todas las hojas e incluir en cada una el nombre, la cédula de identidad, el número de hoja y el TOTAL de hojas.
- Utilizar las hojas de un solo lado y escribir con lápiz, iniciando cada ejercicio en hoja nueva.
- En la **carátula**: completar el índice indicando en qué hoja se comenzó la respuesta a cada problema/ejercicio y el total de hojas entregadas.

## Parte Obligatoria (Ejercicios 1 a 4)

Esta parte es eliminatoria, para la aprobación del examen debe obtenerse un mínimo del **50% de esta parte (20 puntos)**. En caso de no llegar a dicho mínimo, **NO** se corregirán los problemas.

**Ejercicio 1** En este ejercicio tomamos como operación básica el acceso a los arreglos. Considere el siguiente programa:

```
int pgm (int * A, int n, int s) {
    int cnt = 0;
    for (i = 0; i < n; i++)
        for (j = i+1; j < n; j++)
            if (A[i] + A[j] == s) cnt ++;
    return cnt;
}
```

1. Indique qué computa el programa `pgm`.
2. Calcule la cantidad exacta de operaciones básicas  $T(n)$  que realiza este algoritmo.
3. Indique si las siguientes oraciones son verdaderas o falsas. Justifique.

a)  $T(n) \in O(n \times (n - 1))$

b)  $T(n) \in O(\log n)$

c)  $T(n) \in \Theta(\frac{1}{2}n^2)$

**Ejercicio 2** El contrario de las reglas del límite no siempre valen. En particular, demuestre que la siguiente oración es falsa:

Sean dos funciones  $f : \mathbb{N} \rightarrow \mathbb{R}^+$ ,  $g : \mathbb{N} \rightarrow \mathbb{R}^+$ . Si  $f \in O(g)$  y  $g \in O(f)$ , entonces  $\lim_n \frac{f(n)}{g(n)} \in \mathbb{R}^+$ .

### Ejercicio 3

Se desea interconectar un conjunto de edificios con tendido eléctrico de manera de garantizar una red de cubrimiento mínimo. El costo del tendido de un edificio a otro es proporcional a la distancia entre ellos así como el tiempo que demora su armado (1 tendido a la vez). Por esto último, se desea garantizar que cada tendido que se agregue a la red asegure la existencia de una única componente conexa (entre edificios). Asumiendo que cuenta con la matriz  $distancia[i,j]$  que indica la distancia entre los edificios  $i$  y  $j$ :

1. Escriba el pseudocódigo del algoritmo entre Prim o Kruskal que utilizaría para cumplir con los requisitos.
2. Indique qué técnica aplica su algoritmo entre las vistas en el curso.

### Ejercicio 4

1. Escriba el pseudocódigo de un algoritmo de sorting que utilice Divide & Conquer.
2. Indique su orden en el mejor caso, peor caso y caso promedio para una entrada de  $N$  elementos.

# Problemas

## Problema 1 (30 puntos)

Una empresa desea realizar una actividad de integración entre sus empleados por lo que le es de interés formar grupos de empleados que usualmente no trabajen juntos. Para cada empleado se conocen los empleados con los que usualmente trabaja y los que no.

El objetivo es averiguar si se puede particionar a los empleados en 2 grupos, de manera que los miembros de un grupo no trabajen usualmente entre sí, aunque cada miembro de un grupo sí lo haga con TODOS los miembros del otro grupo. En caso de ser esto posible se desea determinar los miembros de cada grupo.

1. Modele el problema en términos de grafos, estableciendo claramente la propiedad que debe satisfacer el grafo.
2. **Escriba un algoritmo basado en BFS** que determine si es posible armar dichos grupos, y en caso afirmativo devuelva los integrantes de cada uno.

Considere la siguiente firma para la implementación:

```
bool armarGrupos (Grafo G, int n, ListaVertices &grupo1, ListaVertices &grupo2);  
  
/* El grafo 'G' tiene 'n' vértices identificados de 1 a 'n'. Si el algoritmo  
devuelve false, es irrelevante el contenido de grupo1 y grupo2 */
```

Asuma que:

- Están disponibles todas las operaciones de los TADs vistos en el curso. En particular,  $ListaAdyacentes(G, v)$  devuelve la lista de vértices de  $G$  adyacentes al vértice  $v$ .
  - Las listas de vértices  $grupo1$  y  $grupo2$  están inicializadas como vacías.
3. Indique y justifique la relación que debe existir entre el número de aristas  $a$  y el número de vértices  $n$  del grafo  $G$  para tener el peor caso del orden de ejecución del algoritmo implementado.
  4. Calcule dicho orden en función de  $n$ .

**Nota:** Puede asumir para las partes 3 y 4 que  $n$  es múltiplo de 2.

## Problema 2 (30 puntos)

Una pareja decide tomarse un crucero para su licencia contando con un presupuesto  $P$  que no pueden superar. Existen  $C$  cruceros disponibles, donde cada crucero sale y retorna a una ciudad visitando distintos lugares en su recorrido.

Para tomarse el crucero la pareja debe de viajar hasta la ciudad de donde parte el mismo y luego retornar a su ciudad, no hay cruceros que partan de la ciudad donde vive la pareja, ni que lleguen a ella. Existen  $V$  vuelos en las fechas de la licencia, cada vuelo parte de una ciudad origen y llega a una ciudad destino. La pareja está dispuesta a hacer un máximo de 3 escalas tanto a la ida como a la vuelta, prefiriendo la opción con la menor cantidad de escalas si lo permite el presupuesto.

La pareja debe de ser capaz de llegar a tiempo para la partida de su crucero y los respectivos vuelos.

La pareja designó puntos de preferencia a cada uno de los posibles lugares que podrían visitar con el crucero. Formalice en términos de **Backtracking** y en lenguaje natural la solución que logre sumar la mayor cantidad de puntos según la preferencia de los lugares. Luego considere, para el armado del paquete completo, la preferencia en cuanto a los vuelos de la pareja.

No se deben de plantear los predicados de poda del problema.

Considere los siguientes datos para la solución:

- $pareja.ciudadResidencia$  indica la ciudad donde vive la pareja.
- $puntosPreferencia[k]$  indica la cantidad de puntos que la pareja le otorgó al destino identificado como ' $k$ '.

- *vuelos[i]* refiere al vuelo identificado como 'i'.
- *cruceros[j]* refiere al crucero identificado como 'j'.
- Tanto los vuelos como los cruceros tienen los siguientes campos:
  - o *precio* indica el precio.
  - o *origen* indica la ciudad de origen.
  - o *destino* indica la ciudad destino.
  - o *partida* indica la fecha y hora de partida.
  - o *llegada* indica la fecha y hora de llegada.
- Adicionalmente el crucero tiene los siguientes campos:
  - o *cantLugares* indica la cantidad de lugares que visita el crucero.
  - o *lugares[h]* es el identificador del lugar que el crucero visita en el lugar 'h'.

\*Las fechas y horarios se consideran todos en el mismo huso horario y se pueden comparar con los operadores convencionales.