

EVALUACIÓN 2 – CÁTEDRA
FACULTAD DE INGENIERÍA
Departamento de Ingenierías Multidisciplinares
Métodos de Programación



INSTRUCCIONES GENERALES

1. Escriba el número de su Cédula de Identidad y nombre completo. Indique además la sección a la que pertenece.
2. El estudiante que sea sorprendido en actos deshonestos será calificado con la nota mínima (1,0).
3. La evaluación posee una exigencia del 50%, lo cual genera la escala de notas mostrada en la Figura 1.

Puntaje	Nota	Puntaje	Nota	Puntaje	Nota	Puntaje	Nota	Puntaje	Nota
0.0	1.0	10.0	1.7	20.0	2.3	30.0	3.0	40.0	3.7
1.0	1.1	11.0	1.7	21.0	2.4	31.0	3.1	41.0	3.7
2.0	1.1	12.0	1.8	22.0	2.5	32.0	3.1	42.0	3.8
3.0	1.2	13.0	1.9	23.0	2.5	33.0	3.2	43.0	3.9
4.0	1.3	14.0	1.9	24.0	2.6	34.0	3.3	44.0	3.9
5.0	1.3	15.0	2.0	25.0	2.7	35.0	3.3	45.0	4.0
6.0	1.4	16.0	2.1	26.0	2.7	36.0	3.4	46.0	4.1
7.0	1.5	17.0	2.1	27.0	2.8	37.0	3.5	47.0	4.1
8.0	1.5	18.0	2.2	28.0	2.9	38.0	3.5	48.0	4.2
9.0	1.6	19.0	2.3	29.0	2.9	39.0	3.6	49.0	4.3

Puntaje	Nota	Puntaje	Nota	Puntaje	Nota	Puntaje	Nota	Puntaje	Nota
50.0	4.3	60.0	5.0	70.0	5.7	80.0	6.3	90.0	7.0
51.0	4.4	61.0	5.1	71.0	5.7	81.0	6.4		
52.0	4.5	62.0	5.1	72.0	5.8	82.0	6.5		
53.0	4.5	63.0	5.2	73.0	5.9	83.0	6.5		
54.0	4.6	64.0	5.3	74.0	5.9	84.0	6.6		
55.0	4.7	65.0	5.3	75.0	6.0	85.0	6.7		
56.0	4.7	66.0	5.4	76.0	6.1	86.0	6.7		
57.0	4.8	67.0	5.5	77.0	6.1	87.0	6.8		
58.0	4.9	68.0	5.5	78.0	6.2	88.0	6.9		
59.0	4.9	69.0	5.6	79.0	6.3	89.0	6.9		

Figura 1: Escala de notas .

4. Dentro de los aspectos que se considerarán en la calificación se tiene:
 - ✓ Respuesta a lo solicitado, indicando una respuesta clara y coherente a la pregunta realizada.
 - ✓ Completitud, que la respuesta sea completa y no hayan faltado ítems por ver en ella.
 - ✓ Redacción, donde la lectura sea fluida y no de paso a problemas de entendimiento por parte del lector.
 - ✓ Ortografía, cuidar de la escritura de la respuesta.
 - ✓ Orden, donde claramente se pueda ver la respuesta a una pregunta, sin que dos preguntas sean respondidas en el mismo sector.



5. *Para la pregunta abierta, procure incluir una sección para descripción del problema y otra para la solución. Puede ser tipo informe o bien puede responder utilizando este enunciado incluyendo únicamente sus datos y su respuesta a la pregunta 3, todo lo demás elimínelo.*
6. *La entrega se debe realizar en la plataforma <https://www.udesantiagoovirtual.cl/>, en los links que se han habilitado para cada una de las preguntas y enviado una copia al mail de su profesor de cátedra.*
7. *El no subir una parte de la prueba, o subirla equivocadamente, esta se considerará que no ha respondido la pregunta respectiva*

Consideraciones generales:

- Fecha y hora de inicio de la evaluación: Martes 28/12/2021 08:00 horas.
- Fecha y hora de término de la evaluación: Martes 04/01/2022 a las 23:55.
 - Atrasos: Cualquier atraso en la entrega considerará un descuento de 0.5 décimas a la nota final por cada 30 minutos de atraso.
 - Problemas de entrega: En caso de tener un problema para entregar la evaluación, este debe ser justificado informando al coordinador del curso y a su profesor de cátedra¹. Es posible que, acorde al problema, se le solicite mayor antecedentes o justificación mediante certificados médicos o certificados de la asistente social de la carrera.
- Formato de entrega: Se deberá entregar en un archivo comprimido el código correspondiente a su respuesta. El nombre del archivo comprimido debe cumplir con el formato: <RUN SIN DV>_<APELLIDO>_<NOMBRE>_<SECCIÓN>.<EXTENSIÓN>, de esta forma, si mi RUN es 15.324.764-1, mi nombre es Frank Jhon Castle Bernthal y mi sección es la D-5, subo mi archivo en un .rar, entonces el nombre de archivo debe ser: 15324764_Castle_Bernthal_Frank_Jhon_D5.rar.
- Para el desarrollo de la evaluación utilice el lenguaje de programación C junto con el compilador gcc.
- Recuerde utilizar buenas prácticas de programación y comentar su código para las preguntas uno y dos.

¹ La información de contacto del curso está en el documento: Directorio Docentes y Ayudantes disponible en el curso de Moodle.

8.

PREGUNTA 1 (30 puntos):

Las redes de computadores han estado en riesgo constante debido al surgimiento de nuevos gusanos informáticos. Estos son programas que logran transmitirse en la red a otros computadores replicándose. Recientemente un nuevo gusano variante de MyDoom² contagia entre 2 a 4 computadores, lo cual es una tasa baja en relación a sus antecesores pero esto lo hace menos detectable y puede bloquear totalmente al computador.

En el día de hoy es importante saber en qué momento se puede ver colapsado las redes de computadores de un país en específico, para lo cual se considerará lo siguiente:

1. Se considerarán cuatro tipos de computadores:
 1. No infectados: Computadores sin el gusano, pero que se pueden contagiar en cualquier momento.
 2. Infectados: Computadores infectados, pero estos NO transmiten la enfermedad.
 3. Contagiantes: Computadores infectados que transmitirán el gusano.
 4. Bloqueados: Computadores infectados que han sido bloqueados por el gusano.
2. Cada computador contagiado contagiará a 5 computadoras a la semana.
3. Los computadores contagiados podrán contagiar a otros computadores en una semana más después de contagiarse, de esta forma, si un computador se contagia en la semana 4, a la 5 se considerará computador contagiado, pero a la semana 6 se considerará contagiante.
4. Después de una semana, se bloqueará (dejará de funcionar) el 2.5% de los contagiantes.
5. El sistema de seguridad de redes es capaz de mantener un 0.01% del total de computadores del país.
6. Los sistemas de seguridad de redes del país aumentarán su capacidad en un 5% semanal después de que la suma de los contagiados y contagiantes superen el 10% de la cantidad de computadoras total del país.

² <https://es.wikipedia.org/wiki/MyDoom>

7. Los sistemas de seguridad del país se verán colapsados cuando la suma de los contagiados y contagiantes superen el 60% del total de computadores del país.

Construya un programa, que, dado la cantidad inicial de contagiantes y una cantidad total de computadoras del país, sea capaz de indicar en cuántas semanas el sistema de seguridad del país se verá colapsado.

De esta forma, se tendrá la siguiente evolución a través del tiempo el ejemplo mostrado en la tabla N°1, la semana N°16 es dónde el sistema colapsa.

N° Semana	Computadores Infectados	Computadores Contagiantes	Bloqueados	Computadoras libres	Computadores Totales	Capacidad Seguridad	Sobrepasa el 10%	Colapsa el sistema
Semana 0	-	3	-	99.999.997	100.000.000	10.000	NO	NO
Semana 1	15	3	-	99.999.982	100.000.000	10.000	NO	NO
Semana 2	15	18	-	99.999.967	100.000.000	10.000	NO	NO
Semana 3	90	33	1	99.999.876	100.000.000	10.000	NO	NO
Semana 4	165	123	4	99.999.708	100.000.000	10.000	NO	NO
Semana 5	615	288	11	99.999.086	100.000.000	10.000	NO	NO
Semana 6	1.440	903	34	99.997.623	100.000.000	10.000	NO	NO
Semana 7	4.515	2.343	93	99.993.049	100.000.000	10.000	NO	NO
Semana 8	11.715	6.858	264	99.981.163	100.000.000	10.000	NO	NO
Semana 9	34.290	18.573	728	99.946.409	100.000.000	10.000	NO	NO
Semana 10	92.865	52.863	2.050	99.852.222	100.000.000	10.000	NO	NO
Semana 11	264.315	145.728	5.693	99.584.264	100.000.000	10.000	NO	NO
Semana 12	728.640	410.043	15.944	98.845.373	100.000.000	10.000	NO	NO
Semana 13	2.050.215	1.138.683	44.411	96.766.691	100.000.000	10.000	NO	NO
Semana 14	5.693.415	3.188.898	124.133	90.993.554	100.000.000	10.000	NO	NO
Semana 15	15.944.490	8.882.313	346.191	74.827.006	100.000.000	10.000	SI	NO
Semana 16	44.411.565	24.826.803	966.861	29.794.771	100.000.000	10.500	SI	SI

Tabla N°1: Ejemplo de entrada: Total computadores = 100.000.000 y una cantidad de infectados iniciales de 3.

PREGUNTA 2 (30 puntos):

Se dispone de un arreglo **A** de n elementos con las preferencias personales de un grupo de personas. Dichas preferencias se numeran de 1 a k . Por ejemplo para $n=5$ personas y $k=3$ preferencias, se tiene el ejemplo: **A**=[1,3,3,3,2], o **A**=[1,3,3,2,2]. Esto servirá para poder ofrecer productos más preferidos en sitios web.

Se requiere desarrolle un algoritmo recursivo que encuentre el elemento mayoritario dentro del arreglo o indicar con un valor "0" que no existe dicho elemento. El elemento mayoritario, que es aquel que se encuentra repetido más de la mitad de las veces (al menos $1+n/2$). En el primer ejemplo anterior el elemento más repetido es "3" y además es mayoritario, en cambio en el segundo existen dos elementos más repetidos pero ninguno es mayoritario. Para ello debe considerar:

- Utilice División y Conquista (D&C) para desarrollar dicho algoritmo. Explique claramente por qué corresponde genuinamente a un algoritmo de D&C (**5 puntos**).
- Realice el código (basado en su algoritmo) utilizando el lenguaje de programación C. (**15 puntos**).
- Debe realizar una descomposición en sub-problemas. Debe descomponer el problema en al menos 3 partes, explique cuál es la ventaja de usar este método y fundamente su diseño. (**10 puntos**)

PREGUNTA 3 (29 puntos):

En base a la carrera que usted se encuentra estudiando, investigue un problema común de su área la cual sea resuelta utilizando **recursión** o **división y conquista**. Se espera que usted describa un problema y explique cómo el método escogido se encuentra presente en la solución del problema. Puede responder aquí mismo o generar un pequeño informe de no más de 5 páginas.

Ejemplo: en ingeniería informática existen una inmensidad de problemas, como clasificar imágenes, reconocer patrones, problemas de regresión lineal, aprendizaje automático, entre otros. Para el caso de reconocer patrones, la solución puede ser desarrollada de diferentes maneras, sin embargo, la recursividad se encuentra presente en la mayoría de las soluciones. Un desarrollo en términos recursivos, podría ser el siguiente:

El algoritmo reduce progresivamente la complejidad del patrón a reconocer, y al llegar al nivel atómico (caso base, donde se tienen patrones base no divisibles), inicia el proceso de reconocimiento de patrones, desde los atómicos hasta el más complejo. Todo ese proceso se resuelve de forma recursiva, tal que a medida que se descompone se invocan los respectivos módulo de reconocimiento de patrones del siguiente nivel, hasta llegar el nivel atómico, el cual regresa sí reconoció. Así, el algoritmo se invoca a sí mismo (cuando realiza solicitudes de reconocimiento, y va regresando los resultados (si reconoció o no a su patrón) de cada llamada recursiva. De esta manera, el caso base de la recursividad es el nivel atómico, y los procesos de descomposición y reconocimiento que se dan en los otros niveles son los casos generales. Además, entre ellos solo se envían señales indicando si se reconoció o no el patrón respectivo. Es decir, los parámetros que se envían es simplemente una señal, diciendo sí reconoció o no ³.

En el ejemplo anterior se puede ver el problema en concreto y una explicación de cómo la recursión es utilizada, por lo tanto, se espera que usted de manera detallada explique el problema y la solución.

³ http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0254-07702017000200005