

Algoritmos y Estructuras de Datos
Examen Final, "it ends tonight".

Conversion de
Notas:
60-69 = 6
70-79 = 7
80-89 = 8
90-96 = 9
97+ = 10

Nombre:

Legajo:

Instrucciones: Los ejercicios deberían poder completarse en el enunciado mismo, en caso de quedarse sin lugar puede usar alguna hoja auxiliar.

Rúbrica:

Ej	1	2	3	4	5	6	7	8	9	10	Total
Pts	10	10	10	10	10	10	10	10	10	10	100

[1] Indicar V o F en cada una de las siguientes afirmaciones justificando de forma minimalista su respuesta, por ejemplo "F, no existen cuervos amarillos".

- Una función de hashing es buena si dado $h(x)$ es muy difícil hallar x .

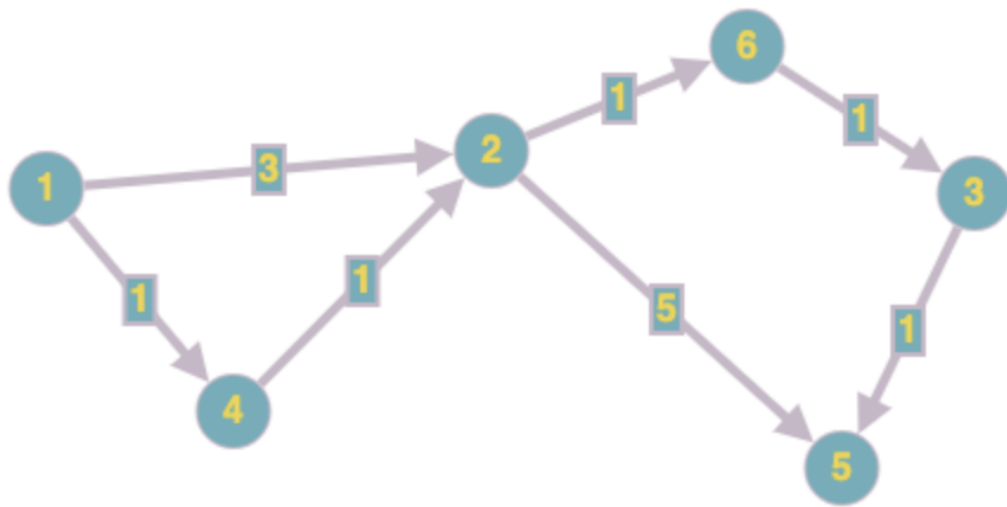
- En una función de hashing criptográfica dado un $h(x)$ todos los x posibles tienen que ser equiprobables. _____
- La construcción de Carter-Wegman permite crear infinitas funciones de hashing a partir de una función de hashing genérica. _____
- Si encontramos x e y tales que $h(x) = h(y)$ entonces la función de hashing no puede ser criptográfica. _____

[2] Realizar las siguientes operaciones sobre la tabla de hashing usando el método del cuckoo con dos tablas. Trate de indicar las operaciones sobre las mismas tablas tachando las claves que son reemplazadas.

Pos	Tabla 1 Clave	Tabla 2 Clave
0		
1		
2		
3		

insert("azul"), insert("verde"), insert("amarillo"), insert("blanco"), insert("naranja"), insert("rojo")
 $h_1(\text{azul})=2$, $h_2(\text{azul})=0$, $h_1(\text{verde})=3$, $h_2(\text{verde})=2$, $h_1(\text{amarillo})=2$, $h_2(\text{amarillo})=1$,
 $h_1(\text{blanco})=2$, $h_2(\text{blanco})=1$, $h_1(\text{naranja})=0$, $h_2(\text{naranja})=1$, $h_1(\text{rojo})=3$, $h_2(\text{rojo})=1$
 Finalmente indique si existe alguna combinación para h_1 y h_2 tal que la próxima inserción falle.

[3] Dado el siguiente grafo:



Calcular todos los caminos mínimos desde el nodo 1 usando el algoritmo de Dijkstra.

Puntitos extras: ¿Funcionaría Dijkstra si en lugar de tener pesos en cada arista tuviéramos un peso/costo en cada vértice del grafo?

[4] Dado un grafo sin ciclos (un árbol) queremos que programe la función `find_best_root(g)` que devuelva el nodo que podría ser la raíz del árbol de altura máxima. No hace falta validar que el grafo sea un árbol. Indique la complejidad del algoritmo que programó. Puntos extras: programar `find_worst_root(g)` que devuelva la raíz a partir de la cuál tendríamos el árbol de altura mínima.

[5] Tenemos un conjunto de monedas de diferentes valores, el objetivo es encontrar de cuántas formas posibles podemos sumar un cierto valor a partir de dicho conjunto. Por ejemplo si nuestras monedas son [1,2,3] para el valor 4 la respuesta es 4 ya que hay 4 combinaciones posibles: (1,1,1,1) (1,1,2) (2,2) (1,3). Notar que (1,3) y (3,1) son repetidos ya que en ambos hay una moneda de 1 y una moneda de 3. Programar una función que calcule a partir de una lista de monedas y un entero la cantidad de combinaciones que existen para sumar dicho entero a partir de las monedas.

[6] Dado el siguiente vector:

11	78	21	51	52	53	44	90	45	9	20	30
----	----	----	----	----	----	----	----	----	---	----	----

Usando bloques con $b=3$ y tablas dispersas tanto para el superbloque como para los bloques individuales le pedimos que muestre la tabla del superbloque y las tablas individuales de cada bloque, estas últimas tiene que construirlas usando el método de los cuatro rusos. Una vez construidas las estructuras indique cómo resuelve las consultas $RMQ(1,7)$. $RMQ(4,11)$

[7] Dibujar el árbol cartesiano para el siguiente vector: 24,38,91,5,16,3,12,15,27
Indique también el número cartesiano correspondiente al mismo.

[8] Tenemos un count-min filter que tiene la siguiente estructura:

0	3	0	2	0	1	1
1	0	0	3	1	1	1
2	0	0	2	2	0	1

- a) Indique qué valores deberían tener las funciones de hashing del string "hola" para que la estimación del mismo sea "2".

- b) Indique cuántos elementos tendríamos que insertar como mínimo para que el siguiente tenga necesariamente una estimación incorrecta (es decir 2 en lugar de 1)

- c) Queremos eliminar del filtro el elemento "gian" para el cual sabemos que $h_1=1$, $h_2=3$, h_3 ¿Cambia la cardinalidad máxima que el filtro puede devolver?

- d) Sabemos que hemos insertado 7 elementos en el filtro. ¿Cuántos elementos distintos puede haber como mínimo?

[9] Dibuje un grafo tal que:

- El grafo debe ser no-dirigido.
- El grafo no puede tener loops (aristas que apuntan a ellas mismas)
- El grafo debe tener al menos 2 ciclos.
- El grafo debe tener al menos 6 aristas
- El diámetro del grafo tiene que ser 4.

Finalmente aplique DFS y BFS sobre su grafo indicando en qué orden se visitarían los nodos.

[10 a] Considerando que solo hay 64 caracteres posibles indique qué longitud deberían tener las claves de forma tal de garantizar que SHA256 tenga al menos una colisión (aunque sea difícil hallarla).

[10 b] Consideremos la construcción de un diccionario en dónde sabemos que el usuario puede hacer “n” inserciones pero solo una consulta. Queremos que tanto insert como search sean $O(1)$. ¿Podríamos usar una lista simplemente enlazada para implementar el diccionario?