**Estructuras de Datos y Algoritmos – iic2133**

**Control 7**

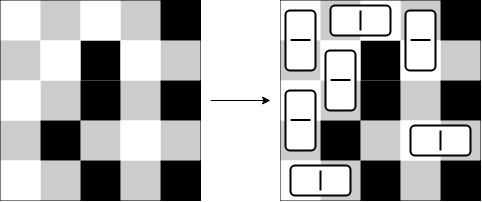
19 de junio, 2019

**Nombre**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1**) Describe un algoritmo que, dados *n* enteros en el rango 0 a *k*, preprocesa el input y luego con-testa cualquier consulta acerca de cuántos de los *n* enteros caen en un rango [*a* ... *b*] en tiempo O(1). Tu algoritmo debe usar un tiempo de preprocesamiento Θ(*n* + *k*).

*Hint*: Recuerda el algoritmo *counting-sort* estudiado en clase que, además de tener un arreglo de entrada y otro de salida, usa internamente un arreglo auxiliar. Más allá de la inicialización de este arreglo auxiliar, *counting-sort* ejecuta tres etapas. La tercera etapa es la que pone los datos del arreglo de entrada ordenadamente en el arreglo de salida; recuerda las dos primeras.

**2**) Se tiene un tablero tipo ajedrez de como el de la figura, en el que algunas celdas (negras) se encuentran bloqueadas; queremos saber la **máxima** **cantidad** de dominós que podemos poner en el tablero para rellenar las celdas que están disponibles (blancas y grises):



Explica **detalladamente** como aplicar flujo máximo en una red de flujo para resolver este problema. Específicamente:

* Explica cómo se construye la red de flujo a partir de un tablero dado.
* Justifica por qué el flujo máximo en esta red corresponde a la cantidad máxima de dominós que es posible poner según las restricciones dadas.