Estructuras de Datos y Algoritmos - IIC2133
Pauta Control 7

1) Describe un algoritmo que, dado *n* enteros en el rango 0 a k, preprocesa el input y luego contesta 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 $\Theta(n + k)$.

   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.

   (4 puntos por preprocesamiento:
   - 1 pto. por crear un array de o a k
   - 1 pto. por contar cuántos de cada número hay
   - 2 ptos. por contar el acumulado del array.)

```
preprocesamiento (data, k, n):
        sea count[0 … k] un nuevo arreglo
        for i = 0 … k:
                count[ i ] = 0
        for j = 1 … n:
                count[data[ j ]] = count[data[ j ]] + 1
        for i = 1 … k:
                count[ i ] = count[ i ] + count[ i - 1]
        return count
```

   (2 puntos por hacer la consulta.)

```
consulta (a, b, count):
        rango = count[ b ] - count[ a - 1 ]
        return rango
```

2)

Mientras expliquen de una forma clara cómo armar el grafo, se consideró correcto. Aquí una propuesta de solución:

Armar un grafo bipartito en que un lado del grafo corresponden a vértices representados como celdas blancas en el tablero y el otro lado corresponda las celdas grises del tablero (también pueden usarse celdas pares e impares según su ubicación dentro del tablero). [1 pto]

Todos los vértices que representan una celda blanca se unen a un único nodo origen del flujo con aristas de costo 1. Todos los vértices que representan una celda gris se unen a un único nodo T, sumidero, con aristas de capacidad 1. No se consideran las celdas negras. [1 pto]

Se puede de esta forma plantear para cada vértice blanco (que representa una celda blanca del tablero) las aristas que representa las celdas grises que se encuentran arriba, abajo, derecha o izquierda del tablero. De esta manera, a partir de un vértice blanco, se genera una arista de capacidad 1 a cada vértice gris. Se puede representar entonces un dominó que se puede colocar entre un vértice blanco y gris, para un problema de flujo máximo. [1 pto]

[3 ptos]

Dado que desde el origen una unidad de flujo puede llegar a cada vértice blanco, esto quiere decir que al menos se puede colocar para cada celda blanca un dominó. A partir de cada vértice se puede pasar esa única unidad de flujo a través de una de las aristas hacia algún nodo gris posible. Dado que para los nodos grises solo se tiene una arista que llega al vértice sumidero, esto obliga a que para una unidad de flujo que pase de un nodo blanco a gris, solo se pueda asignar un único dominó [2 puntos si no se menciona el flujo desde los grises al sumidero]. Se representa a través del flujo máximo la máxima cantidad de dominós que pueden asignarse dado un tablero, dado que representa el problema del tablero y sus restricciones, asignando a lo más 1 dominó por cada celda blanco y gris. [3 ptos]