Matemáticas Discretas 2023-1
Examen 4

Este es un examen de trabajo individual. Debes entregarlo mediante Moodle, con todas las indicaciones precisadas en la actividad correspondiente.

1. (2.5 pts) Considera la gráfica de aristas ponderadas cuyos vértices son los números $1, 2, \ldots, 12$ y en donde dos números i, j tienen una arista entre ellos si i divide a j, o j divide a i. En ese caso, a esa arista se le pone peso $|i-j|$. Usa el algoritmo de Prim manualmente para encontrar un árbol de peso mínimo en esta gráfica.

2. (2.5 pts) Se tiene una gráfica dada G en la cuál se sabe que dos vértices u y v **no** están conectados entre sí, es decir, no hay ningún camino posible de u a v. Queremos agregar a G más aristas, pero cuidando que u y v sigan desconectados.

   Diseña un algoritmo que determine cuál es el máximo número de aristas que se pueden agregar de modo que u y v sigan desconectados. De entrada, recibe una gráfica en formato de vértices y aristas. De salida, debe de regresar tanto el máximo número de aristas que se pueden agregar, como la gráfica con las aristas agregadas, en formato de vértices y aristas. El algoritmo debe correr en tiempo a lo más $O(n^2)$.

3. (2.5 pts) Un *fragmento de ADN* es una palabra que usa únicamente las letras A, C, G y T. Se tienen n fragmentos de ADN de longitud 4. Si se tienen dos fragmentos de ADN, podemos *pegarlos* si pasa alguna de las siguientes dos condiciones:

   - Las dos últimas letras del primero son iguales a las dos primeras letras del segundo.
   - Las tres últimas letras del primero son iguales a las tres primeras letras del segundo.

   Por ejemplo, los fragmentos $ACGG$ y $CGGT$ se pueden pegar para obtener $ACGGT$. Los fragmentos $ACTG$ y $TGAA$ se pueden pegar para obtener $ACTGAA$.

   Dados dos de estos n fragmentos F_1 y F_2, propón en pseudocódigo un algoritmo que determine si se pueden pegar algunos de los n fragmentos para obtener un fragmento (de la longitud que sea) que comience con F_1 y termine con F_2.

Explica las suposiciones de tu modelo y evalúa asintóticamente la complejidad en tiempo de tu propuesta.

4. (2.5 pts) Se tienen n intervalos finitos en $\mathbb{R}$, dados como $[a_i, b_i]$ para cada i en $1, 2, \ldots, n$. A partir de ellos, se crea una gráfica G de la siguiente manera: hay un vértice por cada intervalo y hay una arista si los dos intervalos correspondientes se intersectan. Es sencillo verificar si esta gráfica es la gráfica completa en tiempo $O(n^2)$. Veremos cómo hacerlo en menos tiempo.

   - Escríbe qué desigualdades deben cumplir a_i, b_i, a_j, b_j para que los intervalos $[a_i, b_i]$ y $[a_j, b_j]$ se intersecten.
   - Propón en pseudocódigo un algoritmo que use el inciso anterior para determinar si la gráfica G es completa en tiempo $O(n^2)$.
   - Sea b el mínimo de los b_j. Muestra que todas las parejas de intervalos se intersectan si y sólo si b está en todos los intervalos.
   - Propón en pseudocódigo un algoritmo que use el inciso anterior para determinar si la gráfica G es completa en tiempo $O(n)$.

5. (+2 pts extra) Sea A la matriz de adyacencia de una gráfica G con n vértices. Toma enteros i, j en $\{1, \ldots, n\}$ y un entero positivo k. Demuestra por inducción que la entrada ij de la matriz A^k es igual a la cantidad de caminos que hay del vértice i al vértice j que tengan longitud k. Usa esto para proponer un algoritmo que calcule esta cantidad de caminos y analiza su tiempo de ejecución.