Informe Diego Aarón Moreno Galván

Tarea 7: Mo’s algoritmo para Queries.

Programación eficiente

**Descripción general.**

No logré resolver tal cual el problema de chef con el algoritmo de Mo, sin embargo, a continuación, explico lo que realicé y la modificación al algoritmo de Mo que tuve que hacer para resolver el problema.

**Descripción del programa.**

Para resolver el problema, requeríamos hacer un ordenamiento offline de las queries, por la izquierda los ordenamos por bloques de tamaño , en caso de empate, se ordena por la derecha. Despues, responderíamos la primera querie. La siguiente querie se respondería en base a la anterior, solamente modificando unas cuantas aristas involucradas en la anterior. Si la izquierda de la querie es más pequeña, entonces agregaríamos arista, lo mismo pasa si la derecha es más grande (siempre pasa pues ordenamos). Si la izquierda es más grande lo que tendríamos que hacer es dejar de tomar en cuenta algunas aristas del grafo.

Debido a que se me dificultó el como manejar los grafos, solo contaba con un vector “representantes[]”, en el cual, dos nodos estaban unidos por aristas si tenían el mismo representante. Traté de ver el union-find, pero se me dificultó.

Con lo anterior, era capaz de agregar aristas en el grafo, empero no quitar aristas. Así que, no podía realizar el caso en que la izquierda de la querie sea más grande. Por lo mismo, tuve que modificar el algoritmo de Mo y omitir lo de los bloques. Ordené las queries por lado izquierdo y después por derecho totalmente con quick sort en O(Q). Además también tuve que ordenar las aristas por izq y despues por derecha para encontrar el rango que nos pide cada querie.

Hecho lo anterior, solo respondíamos la primer querie agregando las aristas en dicho rango y respondíamos cuantas componentes hay. Para las siguientes, si solo aumentaba el lado derecho de la querie: agregábamos aristas al grafo y respondíamos. Cuando teníamos que aumentar el lado izq. Entonces borrábamos todo y calculábamos las aristas desde ese número.