

G grafo pesado y conexo, $K \in \mathbb{N}$.

Queremos construir un grafo H que tenga un AG cuya arista más pesada es K . Para construir H podemos cambiar pesos de G .

El costo de construir H es: $\sum_{e \in E(G)} |w_H(e) - w_G(e)|$

Tenemos un algoritmo que construye un AGM en $O(m \alpha^{-1}(m))$.

Diseñar algoritmo que encuentra H con costo de construcción mínimo en $O(m \alpha^{-1}(m))$.

1) Construimos T AGM. $O(m \alpha^{-1}(m))$

2) Revisamos todas las aristas de T .

- Si todas las aristas tienen peso $\leq K$ y hay al menos una arista que pesa exactamente K , $H = G$.
Costo de construcción es 0. $O(m)$
- Si la arista de mayor peso es $< K$, buscamos en todo G la arista con peso más cercano a K . Forzamos esta arista en T generando un ciclo, sacamos cualquier arista del ciclo de las que estaban originalmente en T AGM. Cambiamos el peso de la arista agregada por K para construir H . Es el menor costo de construcción porque cambiamos la arista de G con peso más cercano a K . Existe AG con esta arista por lo explicado anteriormente. $O(m)$

- Si hay al menos una arista con peso $> K$, como T es AGM no podemos reemplazar esas aristas por otras de menor peso pues sino T no sería AGM. Entonces cambiamos el peso de todas las aristas en T con peso $> K$ por K para construir H . $O(m)$