

Selección de algoritmo según el contexto

Escenario	Algoritmo Recomendado	Justificación	Estructuras de Datos	Comentario de Escalabilidad (10x nodos)
1. Despliegue de Fibra Óptica (Urbano) Conectar todos los edificios de un barrio denso minimizando cable.	Prim	Al ser una zona urbana, casi cualquier edificio puede conectarse con cualquier otro (Grafo Denso). Prim evita ordenar la enorme cantidad de aristas.	Montículo de Fibonacci (Fibonacci Heap) + Lista de Adyacencia.	Al multiplicar nodos por 10, las aristas pueden crecer hasta 100 veces (V^2). Prim escala mejor aquí ($O(E+V\log V)$) que Kruskal ordenando aristas.
2. Red Eléctrica Rural Llevar electricidad a pueblos lejanos y dispersos.	Kruskal	Los pueblos están lejos; solo existen pocos caminos viables entre ellos (Grafo Disperso). Es más fácil procesar las pocas conexiones existentes.	Union-Find (Disjoint Set) + Algoritmo de ordenamiento (ej. Merge Sort).	Al ser disperso, si los nodos crecen 10x, las aristas suelen crecer también ~10x (lineal). Kruskal se mantiene muy eficiente ($O(E\log E)$).
3. Diseño de Chips (VLSI) Conectar pines en un circuito integrado microscópico.	Prim	La densidad de pines es altísima y las distancias son euclidianas (todos contra todos).	Matriz de Adyacencia + Arreglo de distancias mínimas (para evitar overhead de punteros).	Crítico. Un aumento de 10x en pines genera una explosión cuadrática de conexiones posibles. Prim es obligatorio para no colapsar la memoria ordenando aristas.

4. Red de Tuberías de Agua (Parque Industrial)Conectar naves industriales a la toma principal.	Kruskal	A menudo las tuberías solo pueden seguir calles predefinidas (limitadas opciones de conexión). El grafo no es completo.	Union-Find con compresión de caminos y unión por rango.	Muy manejable. La estructura Union-Find es casi constante en tiempo de ejecución ($\alpha(V)$), por lo que el cuello de botella es solo ordenar las nuevas tuberías posibles.
5. Clustering de Datos (Machine Learning)Agrupar perfiles de usuarios similares.	Prim	Para encontrar el "esqueleto" de similitud, calculas la distancia de cada usuario contra todos los demás (Grafo completo/muy denso).	Montículo Binario (Binary Heap) o Fibonacci si se requiere máxima eficiencia teórica.	Si pasas de 1.000 a 10.000 usuarios, las comparaciones pasan de 1M a 100M. Prim gestiona esto mejor que intentar ordenar 100M de pares de similitud.