

	Jarnik/Prim	Kruskal	Boruvkas/Sollin
Approach	pick random start node each iteration, add the cheapest edge that won't create a cycle	sort the edges from cheapest to most expensive; each iteration adds cheapest edge that won't make a cycle	do kruskal's algorithm in parallel (each node be a component, keep merging components into an overall tree)
Data Structure Used	priority queue of edges that connect node "in" the tree to one outside the tree	sorted list of edges	sorted list of edges connected to each component
Must Be Able To	maintain the priority queue as nodes/edges were added to the tree, check that next cheapest edge won't create a cycle	check that next cheapest edge won't create a cycle	work and coordinate process in parallel