Parte 1

2	Explica las estructuras de datos usadas y su justificación. En el caso que hayan usado heaps, tienen que explicitar que son la estructura de datos ideal para una cola de prioridad. • 1pt: Usar heaps u otra estructura que permita búsqueda, inserción, eliminación y actualización de forma eficiente (según las complejidades pedidas en el enunciado). • 1pt: Justificar que la estructura es adecuada ya que permite modelar una cola de prioridades.
2	Justifica cómo la estructura de datos usada le permite cumplir con las complejidades solicitadas en los eventos (no considerar MERGE). En el caso de que hayan utilizado heaps, deben explicitar que las operaciones sobre heaps (peek, pop, push y update) cumplen con la complejidad exigida en cada evento. En otro caso, deben justificar su implementación. • 0.5 pts: Por explicar cada operación asociado al contexto de la tarea, junto con su complejidad (explicada por su cuenta o citada de clases).
2	Explica a grandes rasgos la implementación del evento MERGE y por qué cumple con la complejidad exigida. • 1pt: Explica las operaciones utilizadas para llegar al resultado (heapify o ir insertando cada elemento en un heap). • 1pt: Logra la complejidad exigida para construir una EDD con todos los agentes.

Parte 2

4	Explica el algoritmo correspondiente para cada evento, no es necesario calcular matemáticamente la complejidad: • Massive: Explica Quicksort o Heapsort correctamente. (1pt) • K-Massive: Explicar correctamente el algoritmo para ordenar arrays k-ordenados (Usar Heap de tamaño K+1 e ir ordenando de a uno, iterando hacia la derecha). (1pt) • Light: Insertionsort cumple con la complejidad. Explicar que es posible porque la cantidad de clones entrantes está acotada superiormente. (1pt) • Total-Order: Explica correctamente algoritmo que utiliza un heap con tamaño |`lugares`| (1pt) Por cada evento, en caso de errores menores, 0.5 pts. Cualquier error significativo (con implicancias en la complejidad) 0 pts. En caso de sugerir un algoritmo que no cumple con la complejidad 0 pts.
2	Se justifica que ya no se puede cumplir con la complejidad pedida ya que no sirve armar el heap con el primer elemento de cada array. Se propone Heapsort, Mergesort o Quicksort. • 2pts: Explicación correcta de que no se puede cumplir y el porqué. • 1pt: Se dice que no se puede mantener la complejidad, pero la explicación no es correcta. O se sugiere y explica correctamente un algoritmo que no se adapta tanto al caso, como Insertion sort o Selection sort • 0 pts: Dicen que sí se puede mantener la complejidad.

Importante: *Corresponde puntaje completo en caso de haber utilizado un algoritmo distinto al indicado en la pauta, siempre y cuando cumpla con la complejidad temporal y espacial exigida.*