

Q₁estionari sobre *threads* i concurrència

1. Explica quins avantatges i desavantatges té el paradigma de programació paral·lela (memòria compartida) en front al paradigma de programació distribuïda (pas de missatges).
2. Escriu una classe en Java que permeti a dos *threads* fer una cerca paral·lela d'un enter a dins d'una **LinkedList**. Concretament, un *thread* ha de començar a fer la cerca pel principi de la **LinkedList** i l'altre pel final.
3. Escriu una classe en Java que permeti una cerca paral·lela d'un enter a dins d'un array. Concretament, la classe dóna aquest mètode:

```
public static int cercaParallela(int aBuscar, int[] Array, int NumThreads)
```

Aquest mètode crea tants *threads* com especifica **NumThreads**, divideix l'array **Array** en tantes parts com **NumThreads**, i dóna a cada *thread* la part de l'array que li correspon per a que faci una cerca seqüencial del valor **aBuscar**. Si un *thread* ha trobat el valor, el mètode retornarà la casella a dins de l'array. Altrament, el mètode retornarà -1.

4. Modifica l'exercici anterior per a que els *threads* accedeixin a l'array via memòria compartida.
5. Justifica quina diferència hi ha entre emprar el mètode **run** i el mètode **start** de la classe **Thread**.
6. Escriu un programa *multithread* en Java que ordena un array recursivament utilitzant el *merge sort*. Així, el *thread* principal crea 2 *threads* i cadascun va creant dos nous *threads* per anar ordenant la meitat de l'array.
7. Compara el temps d'execució de l'exercici anterior amb una ordenació seqüencial. Justifica els resultats.