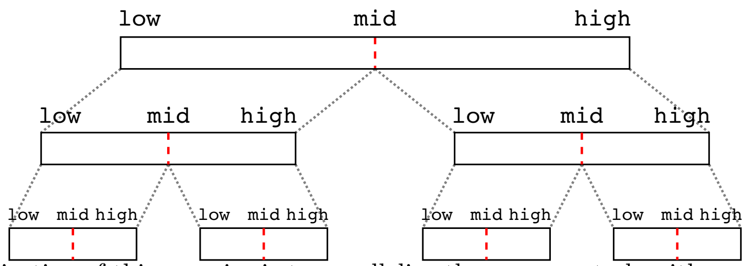
The MergeSort algorithm

Mergesort is a divide and conquer algorithm for sorting the element of an array. It can be implemented as a recursive procedure where the input array is split in two halves, mergesort is recursively called separately on each half and then the results of the two recursive calls are combined together to build the final sorted array.



The objective of this exercise is to parallelize the mergesort algorithm using OpenMP.

1 Package content

The `MergeSort` directory contains a single file named `mergesort.c`. This file contains a main program that allocates an array, fills it with random numbers and then sorts it using the mergesort algorithm. The mergesort algorithm is implemented through the two routines `mergesort` and `merge`; only the first one has to be modified to parallelize the algorithm.

The code can be compiled using the `make` command: simply type `make` inside the `MergeSort` directory. This will generate an executable `main` file that can be launched like this

```
$ ./main n
```

where `n` is the size of the array to be sorted. When executed, the program will print the mergesort execution time or an error message in case the result is not correct.

2 Assignment

- ⌨ parallelize the `mergesort` routine using the `#pragma omp task` construct. Compile and run the `main` program using 1, 2 and 4 threads to verify that the proposed parallelization is correct.

- ✎Evaluate the performance for a large value of `n` (up to 10M) of the parallelization using 1, 2 and 4 threads. Report the measured eecution times in the `responses.txt` file.

3 Advice

- Note that creating and handling tasks has a cost. It may be wise to limit the generation of tasks once a sufficient amount of parallelism has been reached.