1. There is no race condition in this program since each thread can only access unique pairs of elements and there is proper synchronization between even and odd passes.

2. Yes, I think by using n/2 threads that could handle both even and odd comparisons in the same iteration, each thread can perform comparisons for both passes sequentially using good synchronization techniques like barriers to make sure that all threads complete the even pass before any start the odd pass. By doing so, I think it will maintain order and avoid race conditions to make sure that the sorting process is both efficient and correct.

3. Yes, I think that creating n/2 threads at the beginning and also using them throughout the sorting process will have a better overall efficiency since each thread will repeatedly perform its designated comparisons for both passes across all iterations. But to make sure that we stay in good practice of synchronization, we need techniques like barrier at the end of each full iteration after both passes to make sure that all threads are ready to proceed to the next iteration together and so maintaining a correct sort.