Task 1: Advanced Data Structures

Here is the background information on your task

The Walmart Shipping Department is implementing a new system which depends on a priority queue. Since the queue represents a potential bottleneck in the system, much thought is being put into selecting the right data structure for the job. Many options have been proposed, and the plan is to benchmark each one in order to pick the best option. Your task is to implement one of the proposed data structures: a slightly modified heap. The heap must satisfy the heap property, but rather than a traditional binary heap, each parent node in this heap will have 2^x children. Since the goal is to benchmark the most common operations on the data structure, you will only be responsible for implementing two methods - insert, and pop max. Think carefully about what parts of the heap need to change, and how those changes will affect the rest of the heap's behaviour. Good luck!

Here is your task

Your task is to implement a novel data structure - your project lead is calling it a power of two max heap. The rest of your team is doing their best to come up with a better name. The requirements of the data structure are as follows:

- The heap must satisfy the heap property.
- Every parent node in the heap must have 2^x children.
- The value of x must be a parameter of the heap's constructor.
- The heap must implement an insert method.
- The heap must implement a pop max method.
- The heap must be implemented in Java.
- The heap must be performant.
- You must use a more descriptive variable name than x in your implementation.

Think carefully about how you implement each method, and manage the underlying data. Performance is critical, so keep cycles and memory usage to a minimum. Be sure to test your heap with very small and very large values of x. As always, keep a weather eye out for sneaky edge cases.

My Code Explanation and Implementation:

To implement a Power of Two Max Heap in Java, we need to modify the traditional binary heap to accommodate the requirement that each parent node has 2^x children, where x is a parameter provided during the heap's construction.

Key Points

1. **Heap Property**: The heap must maintain the max-heap property, meaning each parent node is greater than or equal to its children.
2. **Parent-Child Relationship**: Each parent node has 2^x children.
3. **Operations**:
    - **Insert**: Add an element to the heap while maintaining the heap property.
    - **Pop Max**: Remove and return the maximum element from the heap, then reorganize the heap to maintain the heap property.

I have attached my code for implementing the Power of Two Max Heap in a file named **heap.java** in **Task 1** Folder.