

1. ****Insertion into Max Heap and Min Heap:****

- The insertion operation in a binary heap has a time complexity of $O(\log N)$, where N is the number of elements in the heap.
- We insert each element into the Max Heap and Min Heap once in the program. Therefore, the overall time complexity for building both heaps is $O(N \log N)$.

2. ****Extracting from Max Heap and Min Heap:****

- The extraction operation in a binary heap also has an $O(\log N)$ time complexity.
- In the program, we perform N extractions from the Max Heap and Min Heap (for N highest and N lowest exchange rates each). Therefore, the overall time complexity for extracting from both heaps is $O(N \log N)$.

3. ****Finding Maximum Subsequence Sum:****

- The algorithm finding the maximum subsequence sum is linear and iterates through the input sequence once. Each iteration involves constant time operations.
- Therefore, the time complexity for finding the maximum subsequence sum is $O(N)$, where N is the size of the input sequence.

4. ****Calculating Average:****

- The average calculation involves iterating the input sequence once and accumulating the sum. Each iteration involves constant time operations.
- Therefore, the time complexity for calculating the average is $O(N)$, where N is the input sequence size.

5. ****Overall Analysis:****

- The dominant factor in the overall time complexity is the building and extraction from the heaps, each having a time complexity of $O(N \log N)$.
- The finding of the maximum subsequence sum and calculating the average contribute linearly to the overall time complexity but are dominated by the heap operations.

6. ****Space Complexity:****

- The space complexity of the program is dominated by the storage of the input sequence in the vectors, resulting in $O(N)$ space complexity.
- The additional space required for the heaps is also proportional to the input data size but is generally smaller than the input data size.

In summary, the overall time complexity of the program is $O(N \log N)$, where N is the size of the input data. The space complexity is $O(N)$. The dominant factors are the heap operations for finding the N highest and N lowest exchange rates.

Overall, heap operations are the dominating factors in terms of time complexity. Therefore, the total time complexity of the program can be expressed as $O(n \log n) + O(n \log n) + O(n) + O(n) = O(n \log n)$, where n is the number of elements in the dataset.