

ChatGPT's answer on 8/22/2023.

Let's analyze the big-O running time complexity of each of the provided code snippets:

p1.cpp

Big-O running time: E: $O(1 + n)$

Explanation: The time complexity of inserting an element into a map is generally $O(\log n)$. However, the loop runs 'n' times, and in each iteration, it inserts an element into the map. Since the insertion operation has a time complexity of $O(\log n)$, and this operation is performed 'n' times, the overall time complexity of the loop is $O(n \log n)$. Additionally, reading the input and returning have constant time complexity, so the total complexity is $O(1 + n)$, which simplifies to $O(n)$.

Big-O running time: D: $O(n)$

Explanation: Similar to p1.cpp, the loop runs 'n' times. However, in each iteration, the code is inserting an element with the key 'n'. Since the key remains the same throughout the loop, it doesn't change the map size, and each insertion essentially overwrites the previous entry with the same key. As a result, the map will only contain one entry with key 'n' and its corresponding value will be the value of 'i' in the last iteration. So, there's only one actual insertion taking place, and since it's done 'n' times, the overall time complexity is $O(n)$.

p3.cpp

Big-O running time: H: $O(n + n \log n)$

Explanation: In each iteration of the loop, the code performs a find operation in the map, which has an average time complexity of $O(\log n)$. Then, it either inserts a new element (which takes $O(\log n)$ on average) or deletes an element (which takes $O(\log n)$ on average). The loop runs 'n' times, so the time complexity for the find and insert/delete operations combined becomes $O(n \log n + n \log n)$, which simplifies to $O(n \log n)$. Additionally, reading the input and returning have constant time complexity, so the total complexity is $O(n + n \log n)$.

Regarding the question about the number of bytes each .cpp file consumes, I can't determine the exact size of the files without knowing the specifics of your system's environment and compiler settings. The file sizes would depend on factors such as the length of variable names, inclusion of any libraries, comments, and how the source code is compiled and stored.