How offset work to calculate mid in binary search

Saturday, November 23, 2024   4:21 PM

The **offset** in the formula
int mid = start + ((end - start) / 2);
plays a critical role in determining the **middle index** of the range [start, end] without causing integer overflow. Let's break it down in detail:

What is the Offset?
The offset is the value (end - start) / 2.
- It represents **how far** the midpoint is from the start index.
- Instead of directly calculating the midpoint as (start + end) / 2 (which may overflow), the formula computes the distance (end - start) first to keep the values manageable.

How the Offset Works
1. **end - start**:
   This calculates the total length of the range (or the number of steps from start to end).
   - For example, if start = 2 and end = 8, then end - start = 6.
2. **(end - start) / 2**:
   Dividing this length by 2 gives the halfway point **relative to the start**.
   - In the above example, (8 - 2) / 2 = 3. This means the midpoint is 3 steps away from start.
3. **start + offset**:
   Adding this offset back to start gives the actual index of the midpoint.
   - Using the above example, mid = 2 + 3 = 5.

Examples
Example 1: Small range
Let's consider start = 0 and end = 7.
- end - start = 7.
- Offset: (7) / 2 = 3.
- mid = start + offset = 0 + 3 = 3.
Here, the offset tells us that the midpoint is **3 steps from the start**.
Example 2: Odd-sized range
For start = 2 and end = 8:
- end - start = 6.
- Offset: (6) / 2 = 3.
- mid = start + offset = 2 + 3 = 5.
Here, the midpoint is at index 5, which divides the range [2, 8] into two subranges: [2, 5] and [6, 8].
Example 3: Even-sized range
For start = 1 and end = 4:
- end - start = 3.
- Offset: (3) / 2 = 1 (integer division truncates fractional parts).
- mid = start + offset = 1 + 1 = 2.
Here, the midpoint is at index 2, which slightly **biases left** because integer division truncates.

Why Use Offset?
The offset works because it **relies on the relative distance between start and end**. It prevents direct addition of large numbers (start + end) and keeps the calculation safe from integer overflow. By focusing on the range's size (end - start) and dividing it in half, the offset ensures:
1. Accurate calculation of the middle index.

2. Efficient and safe computation.

Visualization

Think of the range [start, end] as a number line:

start ---------------------- mid ---------------------- end

The offset (end - start) / 2 measures how far mid is from start. Adding this offset to start shifts the position from the left endpoint (start) to the midpoint of the range.

For example:

- If start = 0 and end = 6, the range length is 6. The offset is 6 / 2 = 3, so mid = 0 + 3 = 3.
- If start = 10 and end = 16, the range length is 6. The offset is 6 / 2 = 3, so mid = 10 + 3 = 13.

In both cases, the offset correctly identifies the midpoint of the range relative to the start.