Understanding Binary Search Termination Conditions

Vishwas Kalani

September 5, 2023

1 Introduction

Binary search is a fundamental algorithmic technique used to efficiently search for an element in a sorted array or solve optimization problems. Understanding the termination conditions of binary search is crucial for its correct implementation.

In this document, I will discuss my insights into binary search termination conditions and how they relate to problem types (minimization or maximization).

2 Termination Conditions

The basic structure of a binary search algorithm involves a `while` loop with the condition $l \leq r$, where l is the left pointer and r is the right pointer. The key to successful binary search lies in maintaining this condition appropriately.

2.1 Minimization Problem

When solving a minimization problem, we typically set an initial result (`res`) and update it within the `while` loop as we narrow down the search space. Here's how the termination conditions work:

a) If the mid-point (`mid`) represents a better solution, update `res` and set `r = mid - 1`.

b) If the mid-point does not represent a better solution, set `l = mid + 1`.

This ensures that `res` contains the minimum value when the loop terminates, and the final condition will be $l > r$, which ends the loop.

2.2 Maximization Problem

For a maximization problem, the approach is similar, but we update `res` differently:

a) If the mid-point (`mid`) represents a better solution, update `res` and set `l = mid + 1`.

b) If the mid-point does not represent a better solution, set `r = mid - 1`.

Again, we ensure that `res` contains the maximum value when the loop terminates, and the final condition is $l > r$.

2.3 Importance of $l \leq r$

It is important to note that we use the condition $l \leq r$ to avoid missing the cases where $l = r$. If we use $l < r$ as the condition, we might skip checking one of the values, which could lead to incorrect results or infinite loops.

2.4 Conclusion

Understanding the termination conditions of binary search is essential for successfully applying this algorithm to various problem types. By maintaining the conditions $l \leq r$ and updating the result (`res`) appropriately based on the problem type, we can ensure the correctness and efficiency of binary search.

3 Alternate termination condition

3.0.1 Alternate Termination Method for Minimization

An alternate termination condition for minimization is to use $l < r$ instead of $l \leq r$. In this case, the logic is as follows:

a) If the mid-point (`mid`) represents a better solution, update `res` and set `l = mid + 1`, `r = mid`, and `mid = (l + r) / 2`.

b) If the mid-point does not represent a better solution, set `l = mid + 1`, `r = mid`, and `mid = (l + r) / 2`.

In this alternate method, we make sure to update `mid` along with `l` and `r`. At the end, we return `l`. This approach is correct because it still maintains the condition that `res` contains the minimum value, and the loop terminates when $l \geq r$.

3.0.2 Alternate Termination Method for Maximization

An alternate termination condition for maximization is to use $l < r$ instead of $l \leq r$. In this case, the logic is as follows:

a) If the mid-point (`mid`) represents a better solution, update `res` and set `l = mid`, `r = mid - 1`, and `mid = (l + r + 1) / 2`.

b) If the mid-point does not represent a better solution, set `l = mid`, `r = mid - 1`, and `mid = (l + r + 1) / 2`.

In this alternate method, we make sure to update `mid` along with `l` and `r`. At the end, we return `l`. This approach is correct because it still maintains the condition that `res` contains the maximum value, and the loop terminates when $l \geq r$.

4 With $r - l > 1$

We can have this condition as well and similar to the above methods we can change l and r in minimization and maximization problem. The mid is always $(l + r)/2$ in this case and in the end we check for l in minimization problem and for r in maximization problem.