

This Document is from ChatGPT

1. Brute Force Approach

- A **brute force approach** means solving a problem in the **most straightforward way possible** without worrying about efficiency.
- You try **all possible solutions** and then pick the correct one.
- It's usually **easy to implement**, but **slow** for large inputs.

👉 Example:

Check if an array has a pair whose sum is equal to a target.

- **Brute force:** Use two loops and check **every possible pair**.
 - ```
for (int i = 0; i < n; i++) {
 for (int j = i+1; j < n; j++) {
 if (arr[i] + arr[j] == target) {
 return true;
 }
 }
}
```
  - Time complexity:  **$O(n^2)$**
- 

## 2. Optimized Approach

- An **optimized approach** uses **better algorithms, data structures, or logic** to solve the same problem more efficiently.
- You **reduce time and/or space complexity** while still solving the problem correctly.
- Requires more thinking but is **faster** and **scalable**.

👉 Example (same problem above):

Check if an array has a pair with sum = target.

- **Optimized:** Use a **hash set** to check in one pass.
  - ```
unordered_set<int> seen;  
for (int i = 0; i < n; i++) {  
    if (seen.count(target - arr[i])) {  
        return true;  
    }  
    seen.insert(arr[i]);  
}
```
 - Time complexity: **$O(n)$**
 - Space complexity: **$O(n)$**
-

🔑 **Difference in Short:**

- **Brute Force:** Simple, tries all possibilities, slow (bad for big inputs).
 - **Optimized:** Smart, efficient, uses better techniques, fast (good for big inputs).
-

⚡ Rule of thumb in DSA:

- 👉 First, write a **brute force solution** (to ensure correctness).
- 👉 Then, think how to **optimize** (using hashing, sorting, binary search, DP, greedy, etc.).