2491. Divide Players Into Teams of Equal Skill

You are given a positive integer array `skill` of **even** length `n` where `skill[i]` denotes the skill of the `i`th player. Divide the players into `n / 2` teams of size `2` such that the total skill of each team is **equal**.

The **chemistry** of a team is equal to the **product** of the skills of the players on that team.

Return *the sum of the **chemistry** of all the teams, or return* `-1` *if there is no way to divide the players into teams such that the total skill of each team is equal.*

Example 1:

```
Input: skill = [3,2,5,1,3,4]
Output: 22
Explanation:
Divide the players into the following teams: (1, 5), (2, 4), (3, 3), where each team has a total skill of 6.
The sum of the chemistry of all the teams is: 1 * 5 + 2 * 4 + 3 * 3 = 5 + 8 + 9 = 22.
```

Example 2:

```
Input: skill = [3,4]
Output: 12
Explanation:
The two players form a team with a total skill of 7.
The chemistry of the team is 3 * 4 = 12.
```

Example 3:

```
Input: skill = [1,1,2,3]
Output: -1
Explanation:
There is no way to divide the players into teams such that the total skill of each team is equal.
```

Constraints:

- `2 <= skill.length <= 10^5`
- `skill.length` is even.
- `1 <= skill[i] <= 1000`

2491. Divide Players Into Teams of Equal Skill

```
/*
   Brute force
   Time complexity: O(n^2),assume serach in unordered set is O(1)
   Space complexity: O(n)
*/

class Solution {
   public:
      long long dividePlayers(std::vector<int>& skill) {
         int n=skill.size();
         int s=0;
         for(auto& e: skill) s+=e;

         if(2*s%n!=0) return -1;

         std::unordered_set<int> taken;
         int target=2*s/n;
         long long ans=0;
         int cnt=0;
         for(int i=0;i<n-1;++i){
            for(int j=i+1;j<n;++j){
               if(taken.find(i)==taken.end() && taken.find(j)==taken.end() && skill[i]+skill[j]==target){
                  ans+=(skill[i]*skill[j]);
                  taken.insert(i);
                  taken.insert(j);
                  cnt++;
               }
            }
         }
         return cnt<n/2||ans==0?-1:ans;
      }
};
```

2491. Divide Players Into Teams of Equal Skill

```
/*
    Sorting
    Time complexity: O(nlogn)
    Space complexity: O(sorting method)
*/

class Solution {
    public:
        long long dividePlayers(std::vector<int>& skill) {
            std::sort(skill.begin(),skill.end());
            int i=0,j=skill.size()-1,total=skill[0]+skill.back();
            long long ans=0;
            while(i<j){
                if(skill[i]+skill[j]!=total) return -1;
                ans+=(skill[i]*skill[j]);
                i++;
                j--;
            }
            return ans;
        }
};
```