# CSC373 – Problem Set 3

Remember to write your **full name(s)** and **student number(s)** prominently on your submission. To avoid suspicions of plagiarism: at the beginning of your submission, **clearly state any resources (people, print, electronic) outside of your group, the course notes, and the course staff, that you consulted**.

Remember that you are required to submit your problem sets as both LaTeX `.tex` source files and `.pdf` files. There is a 10% penalty on the assignment for failing to submit both the `.tex` and `.pdf`.

---

### Due Oct 10, 2020, 22:00; required files: ps3.pdf, ps3.tex

Answer each question completely, always justifying your claims and reasoning. Your solution will be graded not only on correctness, but also on clarity. Answers that are technically correct that are hard to understand will not receive full marks. Mark values for each question are contained in the [square brackets].

**You may work in groups of up to THREE to complete these questions.**

**Please see the course information sheet for the late submission policy.**

**[15 points]**

You are given an array $S$ of $n$ distinct numbers (not necessarily integers) in sorted order such that all numbers in $S$ are non-negative, i.e. $a \geqslant 0$ for all $a \in S$. You are also given a target value $V$. Your goal is to determine the number of pairs $(a, b)$ such that $a, b \in S$ and

$$a^3 + b^3 + 3ab = V.$$

We will consider $(a, b)$ to be the same pair as $(b, a)$. e.g. Given $S = [4, 6, 7, 9, 11]$, and $V = 685$, the answer is 1 (consider the pair $(6, 7)$), and if $V = 863$, the answer is 0.

Systematically design an algorithm for this problem that has a worst-case time complexity of $O(n)$. Note: you're not allowed to use a formula for solving the cubic equation.

1. **(3 points)** Clearly describe all the subproblems that will be solved by your algorithm, and the values that will be stored in all the data-structure being used by your algorithm.
   **Solution:**

   The sub problem that our algorithm needs to solve at each step will be one of the three cases:

   (a) When the result of the given expression with the current a, b is less than the expected result V, the subproblem in this case will be on the list excluding the first element.

   (b) When the result of the given expression with the current a, b is greater than the expected result V, the subproblem in this case will be on the list excluding the last element.

   (c) When the result of the given expression with the current a, b is equal than the expected result V, the subproblem in this case will be on the list excluding the first and the last element.

   The values that will be stored by the algorithm on every iteration will be the list of pairs that satisfied the expression $a^3 + b^3 + 3ab = V$

2. **(5 points)** Write your algorithm in pseudo-code and analyze its complexity.
   **Solution:** The Algorithm on a list lst of size n is as follows:

(a) Solve $a^3 + b^3 + 3ab$ with a=lst[0] and b=lst[n-1]. There are three possible cases that arise from solving this.

(b) Case1: If the expression result is greater than V, then we remove by first (i.e. smallest) element from lst and go back to Step (a) with this smaller lst

(c) Case2: If the expression result is smaller than V, then we remove the last (largest) element from lst and go back to Step(a) with this smaller lst.

(d) Case3: If the expression result is equal to V, then we append (a,b) to our solution list and remove both a and b and continue to step (a) with this new lst

**Pseudo-Code:**

```
def pairs(lst, V):
    sol = []
    while (len(lst)>0):
        a = lst[0]
        b = lst[len(lst)-1]
        expr = (a**3) + (b**3) + (3 * a * b)
        if expr < V:
            lst = lst[1:]
        elif expr > V:
            lst = lst[:-1]
        else:
            sol.append((a,b))
            lst = lst[1:-1]
    return len(sol)
```

Complexity: The number of iterations done by the while loop will always be lesser than or equal to the length of the original list passed to us as we iterate only when the list has elements (and remove either one or two elements from the list at each step depending on the conditions). Hence the complexity of the program is in O(n).

3. **(7 points)** Give a convincing proof of correctness for your algorithm. Remember that if your algorithm is greedy, you will not get any marks for the previous parts without a proof of correctness.
**Solution:**

To prove the correctness of our algorithm,we have to prove that our loop terminates and gives the correct result.
Proof of Termination:
We see that we have three cases under the loop when the expression is less than V, greater than V or equal to V. It is clear from the program that in each of these cases the size of the list is decreasing, so we can consider len(lst) as our loop variant. Since we see that the variant decreases will eventually hit 0, our loop terminates. Hence the program will terminate.

Proof of Correctness:
Let a = lst[0], b=lst[len(lst)-1] and expr = $a^3 + b^3 + 3ab$

Claim: a=lst[0] is always the smallest element of the list
Proof: Since the list given to us is sorted, the first element will always be smallest. We are never changing the position of elements in the list so this always holds.
Claim: b = lst[len(lst)-1] is always the largest element of our list.

Proof: Since the list given to us is sorted, the last element will always be largest. We are never changing the position of elements in the list so this always holds.

Our algorithm within the loop has three different cases. We have to prove that all three cases hold

Case1: expr < V
In this case we remove the first element from our list, claiming that it cannot form a pair with any other element from the list to make expr==V

Claim: a=lst[0] cannot form a pair with any value in the list such that expr==V

Proof: We can prove this by contradiction. Assume by contradiction there exists a value $k \in lst$ such that $a^3 + k^3 + 3ak == V$ But since we have already proven that b is the largest element in the list, it must mean that $k < b$. Since $k < b$ we have that $a^3 + k^3 + 3ak < a^3 + b^3 + 3ab < V$ Hence such a k cannot exist and a cannot be part of a pair.

Since a cannot be a part of a pair, we pop it out of the list and solve the same problem on our new smaller list lst[1:]

Case2: expr > V
In this case we remove the last element from our list, claiming that it cannot form a pair with any other element from the list to make expr==V

Claim: b=lst[len(lst)-1] cannot form a pair with any value in the list such that expr==V

Proof: We can prove this by contradiction. Assume by contradiction there exists a value $k \in lst$ such that $k^3 + b^3 + 3kb == V$ But since we have already proven that a is the smallest element in the list, it must mean that $k > a$. Since $k > a$ we have that $k^3 + b^3 + 3kb > a^3 + b^3 + 3ab > V$ Hence such a k cannot exist and b cannot be part of a pair.

Since b cannot be a part of a pair, we pop it out of the list and solve the same problem on our new smaller list lst[:-1]

Case3: expr == V
In this case we remove both the first and last elements from the list since they have been added to the solution list of pairs now, and are no longer needed as neither a nor b can form a case with any other value.
Claim: Neither a nor b can form another pair with a value from lst.
Proof:

(a) First we can show that a cannot make another pair with other values from lst to make expr == V. Assume by contradiction there exists a value $k \in lst$ such that $a^3 + k^3 + 3ak == V$ But since we have already proven that b is the largest element in the list, it must mean that $k < b$. Since $k < b$ we have that $a^3 + k^3 + 3ak < a^3 + b^3 + 3ab$ Which means that $a^3 + k^3 + 3ak < V$ because $a^3 + b^3 + 3ab == V$ Hence k does not exist and so a cannot make a pair with any other value than b.

(b) Now we show that b cannot make another pair with any other value from lst. We can prove this by contradiction. Assume by contradiction there exists another value $k \in lst$ such that $k^3 + b^3 + 3kb == V$ But since we have already proven that a is the smallest element in the list and we know that all elements of the list are unique, it must mean that $k > a$. Since $k > a$ we have that $k^3 + b^3 + 3kb > a^3 + b^3 + 3ab$ Which means that $k^3 + b^3 + 3kb > V$ because $a^3 + b^3 + 3ab == V$ Hence k does not exist and so b cannot make a pair with any other value than a.

Since whenever we find a pair that satisfies the equation, we append it to our solution list, the length of the solution list gives us the correct number of pairs.