

(a) a verbal description of your algorithm, (b) a corresponding pseudo code, (c) a sketch of a proof of correctness of your algorithm (i.e., an argument that your algorithm always produces the desired answer), (d) a tight running time estimate for your algorithm, and (e) a brief reasoning behind the running time estimate.

Max Pairs

My algorithm allocates an array for all possible sums according to the equation: $n(n-1)/2$, which is all the possible unique sums of two numbers in a list. Then the algorithm goes through a nested for loop that sums every number, and then traces through the list of sums. Sums are stored in a struct that holds both the value and the number of occurrences. If a sum has already been added to the list its occurrence count is increased. Once the nested for loop is done the sum list is traversed to find the sum with the most occurrences, if there is a tie in occurrences the smaller sum is selected.

Pseudocode

```
struct sum{
    float value;
    int count = 1;
}

sum allSums[] = new sum[n(n-1)/2]
float input[] = {...}

//List all unique sums
for (int num1=0; num1 < input.size; num1++){
    for(int num2=num1 + 1; num2 < input.size; num2++){
        float currentSum = input[num1] + input[num2]
        if( allSums contains sum.value==currentSum ){
            sum.count++;
        }else{
            sum newSum = {currentSum}
            allSums.append( newSum )
        }
    }
}

sum maxPairs = {Infinity}
//Find correct output
for each (currentSum in allSums){
    if(currentSum.count > maxPairs.count || (currentSum.count == maxPairs.count &&
currentSum.value < maxPairs.value) ){
        maxPairs=currentSum;
    }
}

return maxPairs
```

Proof of Correctness

Iterating through every possible sum and keeping a running list of all sums and their number of occurrences, guarantees finding the correct output by simply tracing every element in the sum table.

Time Estimate

~~Due to the size of the sum array this runs in $O(n^2)$~~

Note: I realized just now that because of the nested for loop this probably run in $O(n^4)$