

Due at the start of class Thursday, September 12, 2002.

Problem 1. Consider the following code fragment.

```
for i = 1 to n do
  for j = i to 3*i+1 do
    output 'foobar'
```

Let $T(n)$ denote the number of times 'foobar' is printed as a function of n . Express $T(n)$ as a summation (actually two nested summations), and give a closed-form solution for $T(n)$ (i.e. no summations) by simplifying your summation. Show your work.

Problem 2. You are given as input a sequence of $2n$ integers. Design an algorithm with worst case running time $O(n \log n)$ that partitions the numbers into n pairs with the property that the partition minimizes the maximum sum of a pair. For example, if the numbers are $(1, 9, 5, 3)$ the answer is $(1, 9)$ and $(5, 3)$ with maximum value 10. Write down a short argument that justifies why you think your algorithm works.

Problem 3. The mode of a set of numbers is the number that occurs most frequently in the set. The set $\{4, 3, 6, 4, 3, 4, 1\}$ has mode 4. Give an efficient algorithm to compute the mode of a set. Also give the running time of your algorithm.

Suppose we know that there exists an element that occurs at least $\frac{n}{2} + 1$ times in the set. Give an $O(n)$ algorithm to find the mode.

Problem 4. Prove by induction that

$$\sum_{j=1}^n j^2 = \frac{n(n+1)(2n+1)}{6}.$$