Problem 1. (20 points) Prove or disprove the following conjectures:

(a) $n^{\log c} = O(c^{\log n})$.
(b) $2^{n+1} = O(2^n)$.
(c) $2^{2n} = O(2^n)$.
(d) $T(n) = T(n-1) + n = O(n^2)$.
(e) $T(n) = 16T(n/4) + n^2 = O(n^2)$.

Problem 2. (20 points) Integer division $n/d$ produces quotient $q$ and remainder $r$, and can be implemented using repeated subtraction of $d$ from $n$. Give pseudocode for $\text{DIVIDE}(n, d)$ that uses this method to compute $q$ and $r$. Prove your algorithm correct using a loop invariant.

Problem 3. (20 points) Suppose that you want to output 0 or 1, each with probability $1/2$. At your disposal is a procedure $\text{BIASED-RANDOM}$ that outputs 0 or 1 with probability $p$ and $1-p$ respectively ($0 < p < 1$), but you don’t know what $p$ is. Give an algorithm that uses $\text{BIASED-RANDOM}$ as a subroutine and returns an unbiased answer. State the expected running time of your algorithm as a function of $p$.

Problem 4. (20 points) Use heaps to design an $O(n \log k)$ algorithm to merge $k$ sorted lists into one sorted list, where $n$ is the total number of elements in all input lists. (You don’t need to implement basic heap operations.)

Problem 5. (20 points) Describe an algorithm that, given $n$ integers in the range 0 to $k$, preprocesses its input in $\Theta(n + k)$ time and then answers any query about how many of the $n$ integers fall into a range $[a..b]$ in $O(1)$ time. Be mindful of edge cases. (Recall how counting sort computes the number of elements smaller than $x$ for every input element $x$.)

Turn the page for problem 6.
Problem 6. (20 points) Recall the quicksort algorithm:

```
QUICKSORT(A, p, r)
1  if p < r
2      q = PARTITION(A, p, r)
3      QUICKSORT(A, p, q - 1)
4      QUICKSORT(A, q + 1, r)
```

```
PARTITION(A, p, r)
1  x = A[r]
2  i = p - 1
3  for j = p to r - 1
4      if A[j] ≤ x
5          i = i + 1
6      exchange A[i] with A[j]
7  exchange A[i + 1] with A[r]
8  return i + 1
```

And answer the following questions:

(a) What is the running time of PARTITION?
(b) What is the running time of QUICKSORT when all elements of A have the same value?
(c) What is the running time of QUICKSORT when A contains distinct elements sorted in decreasing order?