1. Find the least integer $k$ such that $f(n)$ is $O(n^k)$ for each of the following functions:

(a) $f(n) = 2n^2 + n^3 \log n$
(b) $f(n) = 3n^5 + (\log n)^4$
(c) $f(n) = (n^4 + n^2 + 1)/(n^4 + 1)$
(d) $f(n) = (n^3 + 5 \log n)/(n^4 + 1)$

Solution: In each case we have to find the least integer $k$ such that $f(n)$ is $O(n^k)$ so there must be constants $c$ and $n_0$ such that $f(n) \leq cn^k$ for $n \geq n_0$.

(a) Since $n^3 \log n$ is not $O(n^3)$ (because the $\log n$ factor grows without bound as $n$ increases), $k = 3$ is too small. On the other hand, certainly $\log n$ grows more slowly than $n$, so $2n^2 + n^3 \log n \leq 2n^4 + n^4 = 3n^4$. Therefore $k = 4$ is the answer, with $c = 3$ and $n_0 = 1$.

(b) The $(\log n)^4$ is insignificant compared to the $n^5$ term, so the answer is $k = 5$. Formally we can take $c = 4$ and $n_0 = 1$.

(c) For large $n$, this fraction is close to 1 (this can be seen by dividing the numerator and the denominator by $n^4$). Therefore the answer is $k = 0$, in other words this function is $O(n^0) = O(1)$. Formally we can write $f(n) \leq 3n^4/n^4 = 3$ for all $n \geq 1$, so we can take $c = 3$ and $n_0 = 1$.

(d) Here the answer is $k = -1$, since for large $n$, $f(n)$ is close to $1/n$. Formally we can write $f(n) \leq 6n^3/n^4 = 6/n$ for all $n \geq 1$, so we can take $c = 6$ and $n_0 = 1$. (20 points)

2. You have 81 quarters and a balance. You know that 80 quarters have the same weight, and one weighs less than the others. Give an algorithm (in pseudocode) to identify the light quarter which uses the balance only 4 times in the worst case.
Solution:

for \( k = 1 \) to 4
   put \( 81/3^k \) quarters on one pan and \( 81/3^k \) quarters on the other pan
   if the two pans weigh the same
      then throw away the coins on the balance
   else if the left pan weighs less than the right pan
      then throw away the coins not on the balance and on the right pan
   else throw away the coins not on the balance and on the left pan

(20 points)

3. Suppose that we are given a sorted array of distinct integers \( A[1, \ldots, n] \)
   and we want to decide whether there is an index \( i \) for which \( A[i] = i \).

   (a) Describe a divide-and-conquer algorithm that solves this problem.
   (b) Use the Master Theorem to estimate the running time of the algorithm you described in part (a). Your algorithm should run in \( O(\log n) \) time.

Solution:

(a) Algorithm: If the array has just one integer, then we check whether \( A[1] = 1 \) with one comparison. Otherwise divide the list into two parts, the first half and the second half, as equally as possible. Consider the largest element \( A[m] \) of the left half. We compare \( A[m] \) with \( m \).
   - If \( A[m] = m \), then the answer is yes and we are done.
   - If \( A[m] > m \), then we can throw away the right half and continue recursively in the left half. Indeed, then for every integer \( k \geq 0 \) using the fact that the integers are distinct and sorted
     \[
     A[m + k] \geq A[m] + k > m + k.
     \]
   - If \( A[m] < m \), then we can throw away the left half and continue recursively in the right half. Indeed, then for every integer \( k \geq 0 \) using the fact that the integers are distinct and sorted
     \[
     A[m - k] \leq A[m] - k < m - k.
     \]
Thus for the number of comparisons we get the following recursion:

\[ T(n) = T\left(\frac{n}{2}\right) + 1, \ T(1) = 1. \]

(b) By the Master Theorem, since we have \( a = 1, \ b = 2, \)

\[ f(n) = 1 = \Theta(n^{\log_2 1}) = \Theta(n^0) = \Theta(1), \]

we get Case 2, and thus \( T(n) = \Theta(\log n), \) as desired. (20 points)

4. Assume that our sample space is the set of permutations of the first \( n \) positive integers \((1, 2, \ldots, n),\) and assume that each permutation is equally likely (we have a uniform random permutation). What is the probability that a random permutation has

(a) \( n(n - 1)/2 \) inversions?

(b) 0 inversions?

(c) exactly 1 inversion?

(d) What is the expected number of inversions in a random permutation?

Solution:

(a) The random permutation has \( n(n - 1)/2 \) inversions exactly when it is sorted in decreasing order. Since exactly one of the \( n! \) permutations is sorted in decreasing order, its probability is \( 1/n! \).

(b) The random permutation has 0 inversions exactly when it is sorted in increasing order, its probability is again \( 1/n! \).

(c) The random permutation has exactly 1 inversion in the following \( (n - 1) \) permutations: \((2, 1, 3, 4, \ldots, n), \ (1, 3, 2, 4, \ldots, n), \ldots, \ (1, 2, 3, \ldots, n, n - 1).\) The probability of drawing one of these permutations is \((n - 1)/n!\).

(d) Let

\[ X_{i,j} = \begin{cases} 
1 & \text{if the pair } (i, j) \text{ is an inversion in the permutation} \\
0 & \text{otherwise} 
\end{cases} \]

Then by the linearity of expectation

\[ E(X) = E\left( \sum_{1 \leq i < j \leq n} X_{i,j} \right) = \sum_{1 \leq i < j \leq n} E(X_{i,j}) = \binom{n}{2} \frac{1}{2} = \frac{n(n-1)}{4}. \]

(20 points)
5. Use indicator random variables to find the expected number of balls that fall into the first bin when $m$ balls are distributed into $n$ bins uniformly at random.

**Solution:** Let

$$X_i = \begin{cases} 
1 & \text{if the } i\text{th ball falls into the first bin} \\
0 & \text{otherwise}
\end{cases}$$

Then by the linearity of expectation

$$E(X) = E\left(\sum_{i=1}^{m} X_i\right) = \sum_{i=1}^{m} E(X_i) = \sum_{i=1}^{m} \frac{1}{n} = \frac{m}{n}. $$

(20 points)