Problem 2

a) Students didn’t do well on this question overall.
   Many students correctly explained the difference between the worst case and worst case expected runtimes, but they didn’t state the runtimes for two cases (Θ(n^2) for the worst case and Θ(n log n) for the worst case expected).

b) The majority of the students after swapping the i^{th} element with the last element, used only bubble-down on the i^{th} element, whereas in some cases, bubble-up might be required. Also, some students applied bubble-down on the i^{th} element directly, without swapping the i^{th} and last elements beforehand.

c) Part c was generally well done. Some students had a mistake in the second round of LSD radix sort. In the second round, in addition to ordering the numbers by the second to last digits, they tried to keep the last digits in order as well.
   Typical incorrect answer:
   1st round: [72381, 80801, 65233, 84268, 04598, 99989]
   2nd round: [80801, 72381, 65233, 84268, 04598, 99989]

d) Many students gave the answer of searching for the initial last element k times.
   This gives the correct runtime for MTF, but is not the worst case performance for Transpose.

e) Many answers did not mention that no duplicates implies a binary (not 3-ary) tree so an algorithm can exactly halve the number of options in decision tree each comparison.
   Some answers used some sort of merge sort analysis, which is not a lower bounding technique.

Problem 3

a) This question was done very well.
   The most common mistake was to bound the equation term by term but use −n^2 as an upper bound for −n. A correct bound would have been zero, or, as in many answers, n^2 (though n^2 is unnecessarily large for something that can be bounded by zero).
   A few answers incorrectly claimed the bounds are valid for n ≥ 0, as opposed to n ≥ 1.
Some answers used the same term to bound the sum of multiple terms without justification (e.g. $n^2 \geq -n$ and $n^2 \geq 3$ and saying $n^2 \geq 3 - n$. It may be true, but the justification is invalid)

In a few cases students only showed one of the upper and lower bounds for theta.

Some students did not explicitly state the final $n_0$, $c_1$ and $c_2$.

Some students did not explain why the term $-n + 3$ can be removed.

A few answers incorrectly claimed the bounds are valid for $n \geq 0$, as opposed to $n \geq 1$.

b) Some people proved big-O rather than little-o, or used a big-O approach of finding a $c > 0$ instead of proving it for all $c$.

Some people came to the conclusion of $\frac{d}{c}$ where $d$ is some constant (generally 13 or 40), generally through incorrect simplifications.

Some people simplified $13n + 27$ to $13n$ and solved the inequality for that, which does not work since it gives a value of $n_0$ that only works for a lower bound of the original question, and is thus too small for $13n + 27$.

Some people gave $c$ in terms of $n$, rather than $n_0$ in terms of $c$.

Some students were not using first principles (i.e. using limits, or showing that $13n + 27 \in O(n)$ and then claiming without proof that $O(n) \Rightarrow o(n \log n)$).

Some students got half-way through: e.g. reached an expression with logarithms but did not express $n_0$ in terms of $c$.

A few answers also tried to bound $13n + 27$ by a quadratic, and then messed up the direction of the inequality, i.e. claiming it is valid for $n > n_0$, when the argument implies $n < n_0$ (and is therefore irrelevant to the proof).

c) Many students did not correctly simplify the summation for $\log i$, instead just ignoring it as being dominated by the $n^2$, or simplifying it incorrectly.

Some students treated the case of $i$ being odd as the upper bound and the case of $i$ being even as the lower bound, giving them different complexities for the upper and lower bounds.

Some students simplified the total runs of the first loop to $n!$.

Some students added an extra summation, summing over all $i$, as well as summing over the odd/even numbers in the two different branches.

Many students did not simplify the final Theta bound, giving multiple terms in their result.

d) 1st Part:

Students could either show the recursion in the form of $T(n)$ or the resulting work at each node in the form of either $\Theta(n)$ or $cn$.

Many students simply put $n$ in the tree
2nd Part:
Many students did not put their answer in the form of an "upper bound" - either using big-O or cn. Also, many students did not recognize that the Θ(n) work at each node scales with n so T(n/3) ⇒ cn/3.

3rd Part:
Some students tried to solve out the recurrence relation instead of using the minimum number of levels × work per level. Also, many students answered the lower bound of T(n) as the number of levels so were out by a factor of n.

e) Some students had difficulty with the definition of the expected value.
Some students didn’t read the question carefully enough and interpreted it to mean that if i is odd then the function will print i asterisks (instead of n).

f) Many students did not find the closed form of the formula for T(n), leaving it as a recurrence relation.
Some students gave a result using order notation, or unspecified constants, rather than an exact formula for the expected number of asterisks.
Some students had trouble handling the for loop in the else block, some did not include it in their recurrence relation, some treated i as a given, rather than finding the result for the expected value of i, some used 1/2 rather than 1/4 as the probability of a given odd number, or used 1/4 as the probability of a given odd number without also including the probability of the number being odd.

Problem 4

• The most common mistake was in trying to implement the key-search/insert/delete via a heap, with the incorrect claim that search and deletion are logarithmic.

• Also quite common was in implementing the key-search/insert/delete by a BST, without specifying that it is self-balancing or AVL.

• For deleteIthinserted, most serious attempts only tried removing the ith element from the whole pool, and not the elements remaining.

• This was not penalized, but many answers provided thorough details on how the search/insert/delete operations were performed on established data structures, despite these details already covered in class. Such details were ignored, so there are many cases of students who wrote a lot of details and earned very little for their answer.

Problem 5

ab) The most common issue was not showing both rotations, or forgetting to include balance factors.
c) Most students correctly did the first rotation.

Some students did not do all the necessary rotations (e.g. just showing the first rotation).
Some students made mistakes when doing the double rotation, such as starting by rotating the twelve into the root.
Some students did not show both steps for the double rotation.
Some students did not include some or all the balance factors.

Problem 6

a) Some students seem to have gotten the indexing confused, (e.g. their notes on the number sequence indicating a tower of height 2 but drawing a tower with two nodes rather than 3 etc.)

Many students did not correctly interpret how many numbers should be included in the count step, or would use the different parity number from step 3 as the starting point for step 1. Sometimes students would also ignore the parity of the initial number and just count the number of consecutive numbers of the same parity, even if they did not match the parity of the initial number.

bc) Some students responded using the probability of a given parity as being $\frac{1}{2}$, rather than $\frac{2}{5}$ for odd and $\frac{3}{5}$ for even, given that the numbers generated are between 1 and 5.

c) Some students tried to find the probability as $1 - P(h = 1) - P(h = 0)$, but for $h = 0$ only considered one of the cases of odd followed by even and even followed by odd.

Some students did not consider the probability of the first number having a given parity, instead just summing the results for the second two numbers (i.e. $(\frac{2}{5})^2 + (\frac{3}{5})^2$ rather than $(\frac{2}{5})^3 + (\frac{3}{5})^3$).

d) Some students thought that the expected height of the skip list was constant.