1. Prove that every tournament contains a Hamiltonian path.

**Solution:** Let $G$ be a tournament. If $G$ contains no vertices (the base case), the empty path is a Hamiltonian path of $G$ since it visits every vertex of $G$ exactly once. (We could also use a tournament of one vertex as a base case, but it makes the inductive step a bit more tedious to describe.)

Now, suppose $G$ contains $n \geq 1$ vertices, and assume inductively (the inductive hypothesis) that every tournament of strictly fewer than $n$ vertices contains a Hamiltonian path. Now (the inductive step), let $v$ be an arbitrary vertex of $G$. Consider the two sets of vertices $N^+(v) = \{ u \mid u \to v \text{ is an edge} \}$ and $N^-(v) = \{ u \mid v \to u \text{ is an edge} \}$. Let $G^+$ and $G^-$ respectively be the subgraphs of $G$ induced by $N^+(v)$ and $N^-(v)$, respectively. Both $G^+$ and $G^-$ contain exactly one directed edge between each pair of vertices, so they are tournaments. They also contain strictly fewer than $n$ vertices. Inductively, they each contain a Hamiltonian path. Let $P^+$ and $P^-$ be the paths for $G^+$ and $G^-$, respectively. If $P^+$ is non-empty, there is a directed edge from its last vertex to $v$. Similarly, there is a directed edge from $v$ to the first vertex of $P^-$ if it exists. Path $P^+ \circ v \circ P^-$ exists in $G$ and is Hamiltonian.

We could also solve the problem using a weak inductive hypothesis, but that solution is (in Kyle’s opinion) more difficult to discover. Here is that proof in case you are curious.

**Solution (Using weak inductive hypothesis):** Let $G$ be a tournament. If $G$ contains no vertices (the base case), the empty path is a Hamiltonian path of $G$ since it visits every vertex of $G$ exactly once. Suppose $G$ has $n \geq 1$ vertices, and assume inductively that any tournament of $n - 1$ vertices contains a Hamiltonian path. Let $v$ be an arbitrary vertex of $G$. If $v$ is the only vertex, then it alone is a Hamiltonian path. Otherwise, $G \setminus v$ contains exactly one directed edge between each pair of vertices, so it is a tournament. There exists a Hamiltonian path $P = \langle u_1, u_2, \ldots, u_{n-1} \rangle$ of $G \setminus v$. If edge $v \to u_1$ exists in $G$, then path $v \circ P$ exists in $G$ and is Hamiltonian. If edge $u_{n-1} \to v$ exists in $G$, then path $P \circ v$ exists in $G$ and is Hamiltonian. If neither case holds, then let $i$ be the greatest index such that edge $u_i \to v$ exists in $G$; $i$ is well defined, and $i < n$. Edge $v \to u_{i+1}$ exists in $G$. Therefore, directed path $\langle u_1, \ldots, u_i, v, u_{i+1}, \ldots, u_{n-1} \rangle$ exists in $G$ and is Hamiltonian.

**Rubric:** 10 points total: 2 points for base case; 3 points for strong inductive hypothesis; 1 point for weak inductive hypothesis unless the rest is perfect, then 3 points; 5 points for inductive step

2. Using $\Theta$-notation, provide asymptotically tight bounds in terms of $n$ that answer each of the following questions.

**Solution:**
(a) $\Theta(n^3)$
(b) $\Theta(n)$
(c) $\Theta(n \log n)$
(d) $\Theta(n^2)$
(e) $\Theta(\sqrt{n})$
(f) $\Theta(\log n)$
(g) $\Theta(n)$
(h) $\Theta(1)$
(i) $\Theta(n^3)$
(j) $\Theta(n)$

Rubric: 10 points total: 1 point, all or nothing, per item. No proofs required.

3. Sort the functions listed below from asymptotically smallest to asymptotically largest, indicating ties if there are any.

Solution:

$$20 \equiv 2 + \sin n \ll \log \log n \ll \log n \ll \sqrt{\log n} \ll \log n \equiv H_{\sqrt{n}} \equiv \ln(5n) \ll$$
$$\log^{\sqrt{2}} n \ll \log^2 n \ll n^{1/1000} \ll \sqrt{n} \ll n \equiv H_{2^n} \ll n \log n \ll n \sqrt{n} \ll$$
$$n^2 \equiv 4^{\log n} \equiv 500n^2 \ll n^4 - (n - 1)^4 \ll 2n^{500} \ll 1.001^n \ll 2^n \ll e^n$$

Rubric: 10 points total: -1 point for each pair of consecutive non-equivalent items with an incorrect comparison; -1/2 point for each pair of consecutive equivalent items listed as non-equivalent; minimum of 0 points. No proofs required.

4. More formally, you are given an array $X[1..n]$. The only method you have to compare elements of $X$ is a procedure $\text{SAME}(x, y)$ that returns $\text{TRUE}$ if elements $x$ and $y$ are equivalent and $\text{FALSE}$ otherwise. Design and analyze an algorithm to output a member of $X$ whose equivalence class contains strictly greater than $n/2$ members. For your analysis, give an asymptotic bound on the number of times your algorithm calls $\text{SAME}$.

Solution: We've only learned one algorithm design technique so far in this class, so let's solve the problem using the divide-and-conquer paradigm. Suppose we divide $X$ into two equal sized groups. The main observation we will use is that the majority equivalence class for $X$ is also the majority equivalence class for at least one of the two groups. However, one of the two groups may not contain a majority group. We will describe an algorithm $\text{MAJORITY}$ that outputs an element in the majority set of $X$ if more than half the elements are equivalent. Otherwise, $\text{MAJORITY}$ will output the sentinel value $\emptyset$. 


MAJORITY($X[1..n]$):
  if $n = 1$, then return $X[1]$
  $m \leftarrow \lceil n/2 \rceil$
  $\ell \leftarrow \text{MAJORITY}(X[1..m])$
  if $\ell \neq \emptyset$
    // Check if $\ell$ is a member of the majority set
    $\text{count} \leftarrow 0$
    for $i \leftarrow 1$ to $n$
      if $\text{SAME}(X[i], \ell)$, then $\text{count} \leftarrow \text{count} + 1$
    if $\text{count} > n/2$, return $\ell$
  $r \leftarrow \text{MAJORITY}(X[m+1..n])$
  if $r \neq \emptyset$
    // Check if $r$ is a member of the majority set
    $\text{count} \leftarrow 0$
    for $i \leftarrow 1$ to $n$
      if $\text{SAME}(X[i], r)$, then $\text{count} \leftarrow \text{count} + 1$
    if $\text{count} > n/2$, return $r$
  There was no majority this time!
  return $\emptyset$

The algorithm is correct for $n = 1$, because the one element is the majority of $X$. Now, suppose $n > 1$, and assume inductively that MAJORITY returns a representative of the majority set if there is a majority set and $\emptyset$ otherwise for all arrays of fewer than $n$ elements.

Suppose more than $n/2$ elements are equivalent in $X$, and let $x$ be a member of this majority equivalence class. If $X[1..m]$ contains more than $m/2$ elements from this equivalence class, then the first recursive call to MAJORITY will return some member $\ell$ by the inductive hypothesis. The higher for loop will then discover the majority of elements in $X$ are the same as $\ell$ and return $\ell$ correctly. Now suppose the first call returns some element outside the majority class for $X$ or it returns $\emptyset$. In this case, at most $m/2$ elements of $x$’s class belong to $X[1..m]$. If $n$ is even, $X[m+1..n]$ contains more than $n/2 - (n/2)/2 = n/4$ elements in $x$’s class, which is more than half the elements of $X[m+1..n]$. If $n$ is odd, then $X[m+1..n]$ contains more than $n/2 - (n/2 - 1)/2 = n/4 + 1/4$ elements in $x$’s class, which is again more than half the elements of $X[m+1..n]$. Either way, the second call to MAJORITY will return a representative of that set by the inductive hypothesis, and the second for loop will verify that representative belongs to the majority set of $X$.

Finally, if there is no majority set in $X$, then neither for loops will succeed in increasing $\text{count}$ enough to return an element. MAJORITY will correctly return $\emptyset$.

For the running time, we observe the for loops perform $\Theta(n)$ operations outside of the recursive calls. In the worst case, we make both recursive calls. Let $T(n)$ be the worst case running time. We have

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n).$$

This recurrence is the same one used for merge sort. We saw in class that the solution to this recurrence is $\Theta(n \log n)$. ■
Rubric: 10 points total: 1 point for base case; 4 points for the rest of the algorithm; 3 points for the proof; 2 points for full analysis, 1 point for just giving the recurrence. A correct $O(n^2)$ solution is worth at most 7 points.