CS 130A: Review I Solutions **Winter 2019**

1. (a) Yes. One way to see this is: $n + \log n + \sqrt{n} \ \leq \ 3n$, for all $n \geq 1$.
   (b) Yes. This follows because $\sum_{i=1}^{\sqrt{n}} i \ = \ \sqrt{n}(\sqrt{n}+1)/2 = (n + \sqrt{n})/2$.

2. Since m is even, there are exactly $m/2$ even-numbered and $m/2$ odd-numbered slots. The prob. that a key hashes into an even-numbered slot is therefore $1/2$. The prob. that all n keys hash into even-numbered slots, and thus none in odd-numbered slots, is $(1/2)^n$.

3. We note that

$$E(X) \ = \ \sum_x x Pr(X = x) \ = \ \sum_{x<a} x Pr(X = x) \ + \ \sum_{x \geq a} x Pr(X = x)$$

$$\geq \ \sum_{x \geq a} x Pr(X = x) \ \geq \ a \sum_{x \geq a} Pr(X = x) \ \geq \ a Pr(X \geq a).$$

4. (a) By the heap-ordering property, for all nodes x, we have $key(parent(x)) < key(x)$. So, the maximum cannot be at a non-leaf node because then the children of that node must have keys larger than the max, which is a contraction of the max.
   (b) Prove this by induction.
   (c) Suppose, for the sake of contradiction, tha an algorithm does not check one of the leaf nodes, say, z. Suppose the max item has value L. Feed the algorithm two nearly identical instances, once with the original input, and once with the value at the leaf z changed to $L + 1$. Since only the value of z is changed and the algorithm did not even check z, the algorithm must return the same answer in both cases, but is clearly incorrect for one of them.

5. (a) PercolateUp in a d-heap takes $O(\log_d N)$, while PercolateDown takes $O(d \log_d N)$. So, the total running time is $O(M \log_d N + dN \log_d N)$.
   (b) $O((M + N) \log_2 N)$.
   (c) $O(M + N^2)$.
   (d) Starting at $d = 2$, increasing d makes PercolateUp cheap, but makes Percolate-Down more expensive. So, we need a choice of $d \geq 2$ for which the two balance out. This occurs when $M = dN$, or $d = M/N$. So, we choose $d = \max(2, M/N)$.