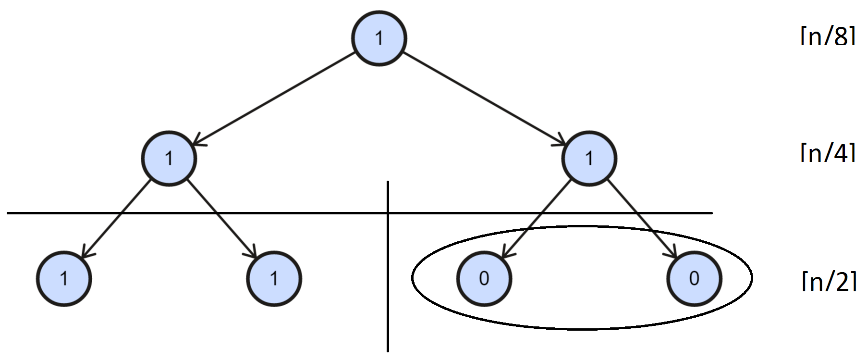
Aditya Shirode, Rhythm Shah, Shrenuj Gandhi

1. [5 points] *Heapsort.*

   Prove that the number of comparisons for Heapsort is $\Omega^\infty(n \lg n)$ in the worst case. You have to show that there exists $c > 0$ such that, for an infinite number of n, there is an instance of size n that causes Heapsort to do at least $cn \lg n$ comparisons. Hint: You can show that this is true even if all the keys are 0's and 1's. The Buildheap stage is not relevant – we have already shown that it takes $O(n)$ comparison – so start with the sorting stage.

Answer:
Let us assume $n = 2^k - 1$, which means that the heap is a complete binary tree.

Consider the case where $k = 3$, which means that the heap has 7 nodes. Also consider a special instance where the right half of the bottom-most layer consists of all 0s.



Here, we are assuming that if both children are equal, then the left child will be swapped (as this will lead maximum comparisons in later stages).
We can say that there are two smaller nodes, i.e. $\frac{1}{2}\lceil\frac{n}{2}\rceil$ that will undergo $2 \lg n$ (height of the tree) comparisons.

Thus,

$$\begin{aligned}\#\text{comparisons in sorting stage of heapsort} &\geq \frac{1}{2}\left\lceil\frac{n}{2}\right\rceil(2\lg n)\\ &\geq \left(\frac{n}{2}+1\right)\lg n\\ &\geq \frac{1}{2}n\lg n + \lg n\end{aligned}$$

This implies that # comparisons done in the sorting stage is $cn \lg n$.
Since k is a positive number, we can state infinitely many instances like this, for which the above case holds true.
Thus, we can say that #comparisons for Heapsort is $\Omega^\infty(n \lg n)$.