A min heap is a (usually) binary tree where each vertex’s data is no larger than its children’s data. Below, we present such a min heap.

Below we present the naive approach to building a heap in two algorithms. It works by repeatedly percolating elements up toward the root to slowly grow the subheap being built.

**Algorithm 1: Heapify(A)**

- **Input:** Array A
- Let $i = 0$
- while $i < |A|$ do
  - PercolateUp(A,$i$)
  - $i = i + 1$
- return
Algorithm 2: PercolateUp($A, i$)

**Input:** Array $A$ and index $0 \leq i < |A|$; the first $i$ elements of $A$ must form a min heap

// Vertex $j$’s children are at $2j + 1$ and $2j + 2$

Let $p = \left\lfloor \frac{i - 1}{2} \right\rfloor$

while $p \geq 0$ and $A[p] > A[i]$ do

\begin{align*}
&\text{Swap } A[p] \text{ and } A[i] \\
&i = p \\
&p = \left\lfloor \frac{i - 1}{2} \right\rfloor
\end{align*}

return

For the purposes of this assignment, you will likely wish to utilize Wolfram Alpha to number crunch sums for you. If you do so, however, do show your work leading up to how you get such a sum.

Problem 1.

a) What is the worst-case runtime of PercolateUp in terms of $i$? Explain your answer.

b) What is the worst-case runtime of Heapify in terms of $n = |A|$? Explain your answer.

c) What is the best-case runtime of PercolateUp in terms of $i$? Explain your answer.

d) What is the best-case runtime of Heapify in terms of $n = |A|$? Explain your answer.

Problem 2.

a) State precisely what PercolateUp($A, i$) does.

b) Assuming PercolateUp is correct, prove Heapify($A$) is correct.

We can build a heap more efficiently. Instead of building it top-down, we can build it bottom-up instead.

Algorithm 3: FastHeapify($A$)

**Input:** Array $A$

Let $i = |A|$

while $i > 0$ do

\begin{align*}
&i = i - 1 \\
&\text{PercolateDown}(A, i)
\end{align*}

return
Algorithm 4: PercolateDown$(A,i)$

**Input:** Array $A$ and index $0 \leq i < |A|$; every element $i < j < |A|$ of $A$ must be the root of a min heap

// Vertex $j$’s children are at $2j + 1$ and $2j + 2$

Let $p = i$
Let $l = 2p + 1$
Let $r = 2p + 2$


// Assume out of bounds values are $\infty$


Assign $p$ to be the index $l$ or $r$ of the swapped element

$l = 2p + 1$
$r = 2p + 2$

Problem 3.

a) What is the best-case runtime of FastHeapify in terms of $n = |A|$? Explain your answer.

b) What is the worst-case runtime of FastHeapify in terms of $n = |A|$? Explain your answer.

Problem 4.

a) State precisely what PercolateDown$(A,i)$ does.

b) Assuming PercolateDown is correct, prove FastHeapify$(A)$ is correct.