Put your name and section number on your solution.

1. Consider an array of size eight with the numbers in the following order 40, 20, 80, 60, 30, 10, 70, 50.
   (a) Form the heap using the standard (Williams) algorithm. Show the heap as a tree. Show the heap as an array. Exactly how many comparisons did heap creation use?
   (b) Start with the heap created in Part (a). Show the array after each element sifts down after heap creation. How many comparisons does each sift use? What is the total number of comparisons after heap creation?

2. We are going to repeat Problem (1) using Floyd’s version for sifting. Consider an array of size eight with the numbers in the following order 40, 20, 80, 60, 30, 10, 70, 50.
   (a) Form the heap using Floyd’s sifting algorithm. Show the heap as an array. Exactly how many comparisons did heap creation use?
   (b) Start with the heap created in Part (a). Show the array after each element sifts down using Floyd’s sifting algorithm after heap creation. How many comparisons does each sift use? What is the total number of comparisons after heap creation?

3. Consider a heap whose size $N$ is large (say at least 351). (It may matter whether $N$ is even or odd.)
   (a) i. Where can the second largest element be in the heap? Briefly justify.
      ii. How can you find it?
      iii. Exactly how many comparisons do you need?
   (b) i. Where can the third largest element be in the heap? Briefly justify.
      ii. How can you find it?
      iii. Exactly how many comparisons do you need?
   (c) i. Where can the second smallest element be in the heap? Briefly justify.
      ii. How can you find it?
      iii. Exactly how many comparisons do you need?
   (d) i. Where can the third smallest element be in the heap? Briefly justify.
      ii. How can you find it?
      iii. Exactly how many comparisons do you need?