Today: *HW03 part 1 due*; binary heaps; §§ 21.*.
Next class: bottom-up splay trees; §§ 22.{1–4}.

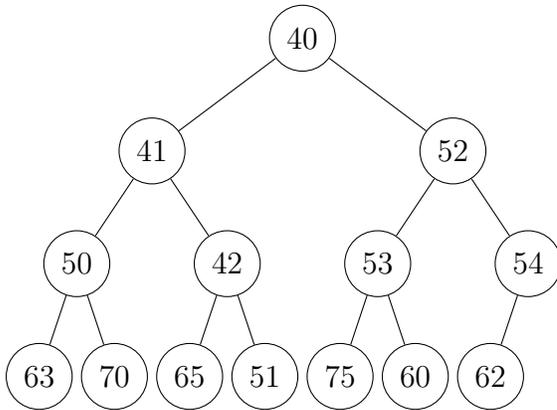1. Write your group identifier (e.g., C3) and its members' names below. Underline your name.

2. Using conventional graphical notation, depict the complete binary tree encoded by the following array, based on the textbook's method.[1]

```
   i:    1    2    3    4    5    6    7    8    9   10   11   12   13   14
a[i]:   50   40   60   70   65   75   62   63   41   42   51   52   53   54
```

3. Mark all violations of the *(min-)heap order property* in the tree of Question 2 by annotating the corresponding edge with a *V*.

---

[1] Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 4th edition (Addison-Wesley, 2010), §21.1.1.

4. Depict the state of the following binary min-heap after all actions triggered by a *deleteMin* operation have completed. Repeat for three additional *deleteMin* operations.

```
                    40
            ┌────────┴────────┐
           41                52
        ┌───┴───┐         ┌───┴───┐
       50      42        53      54
      ┌─┴─┐   ┌─┴─┐     ┌─┴─┐    │
     63  70  65  51    75  60   62
```

5. Starting with the final heap of Question 4, depict the state of the heap after all actions triggered by a *insert(57)* operation have completed. Repeat for operations *insert(33)*, *insert(67)*, and *insert(40)*.

6. *Heapify* the tree of Question 2 using the *buildHeap* operation from the textbook.[2] Depict intermediate states of the tree, including at least the states after buildHeap completes each level of the tree.

[2] *Idem*, §21.3.