CS 240 W2017 Midterm

1. a) F b) F c) F d) T e) F f) T g) F h) ?? i) T j) ?? k) T l) F

2. A) Randomly choosing the pivot makes every case of instance have the same probability, so the best-case and worst-case expected runtimes are the same.

B) We delete A[i] by replacing it with the last leaf in the heap. Then, we compare the replaced A[i] with its parent; if it is larger than its parent, we fix-up, otherwise, we perform fix-down.

C) 45981, 02531, 33001, 33333, 45368, 04598, 45979
33001, 02531, 33333, 45368, 45979, 45981, 04598

D) Didn't do

E) A sequence of k searches on L which is worst case for Transpose but Theta(k + n) for MTF would be alternating between searching for what is initially the last element and the second-last element. For Transpose, this means that these two elements alternate getting transposed k times - we are always searching for the last element, so it is Theta(nk).

For MTF, it would take Theta(n) time to find that element the first time, Theta(n) time to find the other element for search 2, and Theta(1) for the next k-2 searches (since they are moved to the front), so it would take Theta(k+n) time.

F) We can create a decision tree, where each node corresponds to a key comparison. For each child, we continue making key comparisons until we get a result for a sorting permutation. The height of the tree is hence the largest number of key comparisons needed to sort the array. There are n! sorting permutations, and so we have 2^h >= n!.

3. a) Let c = 10 and n0 = 5. We will show that

$$|3n^2 - n + 2| \le |cn^2| \quad , \; n \ge n_0$$

For n >= 5, both functions are positive, so we drop the absolute value signs.

$$3n^2 - n + 2 \le 10n^2$$

$$0 \le 7n^2 + n - 2$$

This is a monotonically increasing function, since every coefficient is positive except the constant, and is greater than 0 for all n >= 5. Hence we have that 3n^2-n+2 <= cn^2 for all n >= n0, and by definition 3n^2-n+2 is in O(n^2).

Let c = 1 and n0 = 5. We will show that

$$|3n^2 - n + 2| \geq c_1|n^2|, \; n \geq n_0$$

For n >= 5, both functions are positive, so we drop the absolute value signs.

$$3n^2 - n + 2 \geq n^2$$

$$2n^2 - n + 2 \geq 0$$

We have that 2(5^2)-5+2 = 47 >= 0. The first derivative is 4n-1, and the second derivative is 4. Both of these are positive for all n >= 5, so the function is monotonically increasing, and thus positive for all n >= 5. Hence we have that 3n^2-n+2 >= c1n^2 for all n >= n0, and by definition 3n^2-n+2 is in Omega(n^2).

Hence it is in Theta(n^2).

b) We prove that for all constants c > 0, there exists a constant n0 >= 0 such that |7n+3| <= c|nlogn|. Since both of these functions are positive for n >= 0, we can drop the absolute value signs. We have that

$$7n + 3 \leq 7n + 3n = 10n$$

$$10n \leq cn\log n$$

$$\frac{10}{c} \leq \log n$$

$$n \geq 2^{\frac{10}{c}}$$

Hence we have that for n >= 2^(10/c), 7n+3 <= cnlogn. Hence 7n+3 is in o(nlogn).

c) i is the number of times the while loop runs. We have that

$$\sum_{k=0}^{i} 2k+1 = j$$

The function runs until the first iteration where j > n. Hence we have

$$\sum_{k=0}^{i-2} 2k+1 \leq n \leq \sum_{k=0}^{i-1} 2k+1$$

$$n \le \sum_{k=0}^{i-1} 2k+1 = i + 2\sum_{k=0}^{i-1} k = i + \frac{2i(i-1)}{2} = i^2$$

$$i \ge \sqrt{n} \in \Omega(\sqrt{n})$$

$$n \ge \sum_{k=0}^{i-2} 2k+1 = i^2 - (2(i-1)+1) = (i-1)^2$$

$$i \le \sqrt{n} + 1 \in O(\sqrt{n})$$

Hence the function is in Theta(sqrt(n))

d) N Y N Y
Y N Y N
Y Y N N
Y Y N N

e) We have that

$$T^{exp}(n) = \frac{5}{6} + \frac{1}{6}n$$

For n=12, the expected number of times an asterisk will be printed is 17/6.

f) $$T(n) = \frac{1}{6}n + \frac{5}{6}(1 + T(n))$$

$$\frac{1}{6}T(n) = \frac{1}{6}n + \frac{5}{6}$$

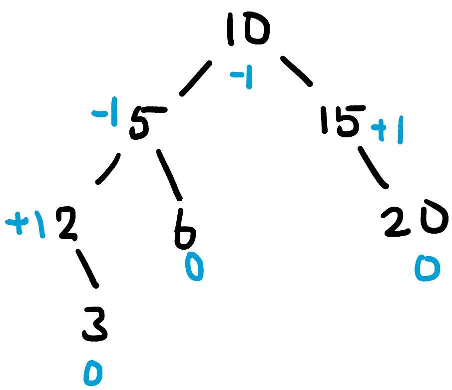
$$T(n) = n + 5$$

4. We can keep a min-heap of size sqrt(n). We first build the heap with the first sqrt(n) elements of the array. This takes O(sqrt(n)) time. We then iterate through the remaining elements of the heap, comparing them to the minimum element of the heap. If the element is greater than the minimum element, we delete that element and replace it with the new element, then perform a fix-down to restore the heap-order property. Each fix-

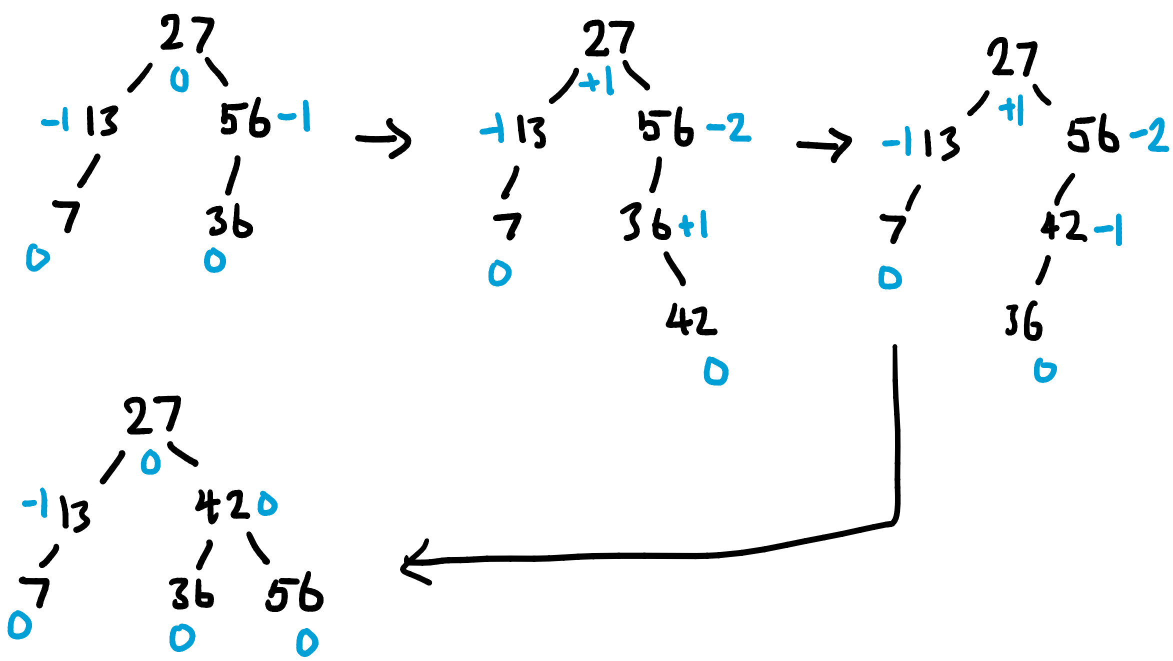
down takes O(log(sqrt(n)) time, and we perform this with at most n-sqrt(n) elements. Hence this step takes O((n-sqrt(n))(log(sqrt(n)) -> O(nlog(sqrt(n)) time. The sqrt(n) largest elements are now all in the heap.

```
heap = MinHeap()
heap.heapify(A[0:sqrt(n)])

for i=sqrt(n) to n
    if (A[i] > heap.root)
        heap.root = A[i]
        heap.fixdown(root)

// sqrt(n) largest elements are now in the heap.
```

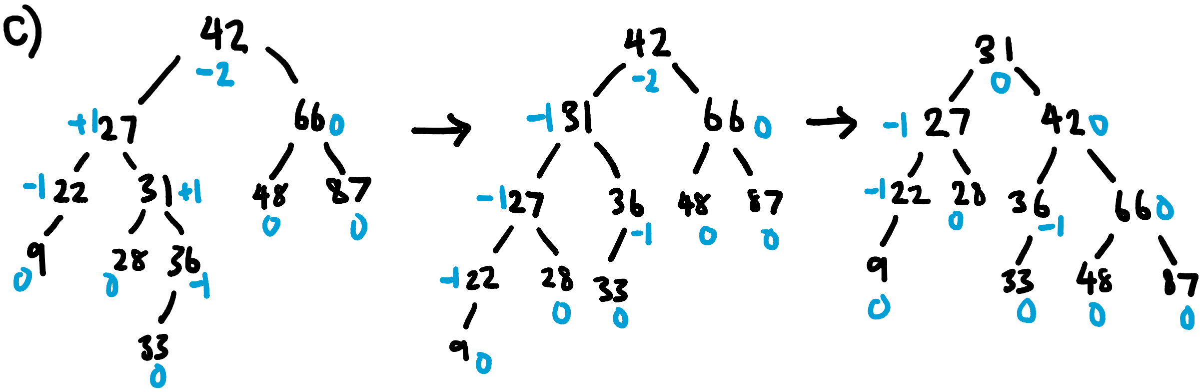
5. a)



b)



c)



6. a)

$-\infty$							∞
$-\infty$					27		∞
$-\infty$	2				27		∞
$-\infty$	2	5			27		∞
$-\infty$	2	5	7	13	27	42	∞

b) For a tower of height two, there needs to be at least a consecutive sequence of two generations of the remembered number, so we have a (1/5)(1/5) = 1/25 chance.

c) The expected height of the skip list is O(log_5(n)).

7. Not covered