

CSE 332: Data Structures and Parallelism

Exercises (Recurrences)

Directions: Submit your solutions on *Canvas*. You must submit a pdf file.

EX11. Recurrences, Recurring (20 points)

```
1 int first17_a(int[] array, int i) {
2     if (i >= array.length) {
3         return -1;
4     }
5     if (array[i] == 17) {
6         return 0;
7     }
8     if (first17_a(array, i + 1) == -1) {
9         return -1;
10    }
11    return 1 + first17_a(array, i+1);
12 }
```

```
1 int first17_b(int[] array, int i) {
2     if (i >= array.length) {
3         return -1;
4     }
5     if (array[i] == 17) {
6         return 0;
7     }
8     int x = first17_b(array, i + 1);
9     if (x == -1) {
10        return -1;
11    }
12    return x + 1;
13 }
```

- (a) [6 Points] What kind of input produces the worst-case running time (number of steps) for `first17_a(arr, 0)`? How about for `first17_b(arr, 0)`?
- (b) [6 Points] Give recurrences (including a base case) for `first17_a` and `first17_b` describing the worst-case running times, where n is the length of the array. You may use whatever constants you wish for constant-time work. Remember that recurrences cannot rely on variables.
- (c) [5 Points] Give a tight $\mathcal{O}(-)$ upper bound for the running time of `first17_a(arr, 0)` and `first17_b(arr, 0)` given your answers to the previous part. That is, find a closed form for your recurrence. Show how you got your answer.
- (d) [3 Points] Give a tight $\Omega(-)$ worst-case lower bound for the problem of finding the first 17 in an unsorted array (not a specific algorithm). Briefly justify your answer.