

Due date: Wednesday, January 25 at 10:30 AM. You will need to submit this via GradeScope. Late problem sets are not accepted.

In CompSci161, your response to each numbered question must be contained within a single piece of paper. Each numbered question must be responded to on a separate page. When you submit to GradeScope, you will need to inform the system which page of your scanned PDF contains the response you want graded. On occasion, we might request you to tag two (or more) parts, even though we know the two parts will be on the same page; when that happens, it usually means we are going to grade the parts separately. Failure to follow these directions will be treated as non-submission for the question(s) affected.

Please review the syllabus and course reference for the expectations of assignments in this class. Remember that problem sets are not online treasure hunts. You are welcome to discuss matters with classmates, but remember the Kenny Loggins rule. Remember that you may not seek help from any source where not all respondents are subject to UC Irvine's academic honesty policy.

1. Consider the problem of finding a *local minimum* in an array. The value $A[i]$ is a local minimum if - and only if - $A[i - 1] \geq A[i] \leq A[i + 1]$. Give an efficient **divide and conquer** algorithm to find a local minimum in an array A in which $A[1] = A[n] = \infty$. State the runtime of your algorithm, both as a recurrence relation and in asymptotic notation and briefly (1-2 sentences is sufficient) explain the correctness of your algorithm.

For example, if your input array A is:

∞	10	23	17	20	1	2	3	4	-3	5	∞
----------	----	----	----	----	---	---	---	---	----	---	----------

then any of the grayed cells could be returned as a valid answer (you need only to find *one* local minimum, not all of them).

A solution with running time $\Omega(n)$ will receive no credit for this problem.

2. The **SORTING HATS** problem is as follows. I have a collection of n hats and n mannequins (statues that resemble a human and are used to model clothing and accessories). My goal is to put one hat on each mannequin head. Each hat is a distinct size, and no two heads are the same size. For each hat, there is exactly one head that the hat fits perfectly. I cannot directly compare two hats or two heads, nor can I inspect a single hat or a single head to determine its size.

I can, however, try to place a hat on a head. If I do so, I find out one of the following pieces of information:

- The hat fits the head perfectly
- The hat is too small for the head
- The hat is too big for the head

We can then remove the hat from the head if we so choose.

The goal of the **SORTING HAT** problem is to match each head with the hat that fits it perfectly while placing as few hats on heads as possible. Give an approach to do so. If there are n

hats and n heads, express the number of times your approach will place a hat on a head in \mathcal{O} -notation and justify why that many such operations will happen. Clearly indicate if you are measuring *worst-case* or *expected case* (either is okay, but be sure to clearly indicate it). Good credit is available for any reasonably-efficient solution, clearly described, with more credit for more efficient solutions.