

Homework 6 problem 2

Problem 2 (10 points):

Suppose A is an array of n integers that is a strictly decreasing sequence, followed by a strictly increase sequence such as $[12, 9, 8, 6, 3, 4, 7, 9, 11]$. Give an $O(\log n)$ algorithm to find the minimum element of the array. Justify your algorithm is correct.

Answer:

algorithm:

```
# l: left pointer
# r: right pointer
# m: medium pointer
# s: input sequence
def find_min(l, r):
    if l < r:
        m = l + (r - l) // 2
        if s[m] < s[m + 1]:
            return find_min(l, m)
        else:
            return find_min(m + 1, r)
    else:
        return s[l]
```

Proof:

Each time the algorithm cut the problem size by half, and by comparing the right element of medium element ($m+1$) to check whether the right side is increasing to decide which side the minimum value not locates in. The algorithm is guarantee to find the minimum value by shrinking the problem size until there is only one integers left, and it is guarantee to halt with one left.

The complexity is $O(\log n)$ since on each recursion level, the algorithm cut the problem by halves, that is $T(n) = T(\frac{n}{2}) = c * \log_2 n = O(\log n)$.