Homework #2
CS 217 @ SJTU
Prof. **Dominik Scheder**
TA. **Tang Shuyang**

Coffee Automaton
ACM Class, Zhiyuan College, SJTU
Due Date: September 29, 2019
Submit Date: September 19, 2019

Exercise 1

The algorithm is described as follows.

1. Make the array into $\frac{n}{2}$ pairs.
2. Sort each of the pair (makes $\frac{n}{2}$ comparisons).
3. For each of the pair, take the smaller one to the S group, and the larger one to the L group.
4. Find the smallest in the S group and the largest in the L group as the answer (each makes $\frac{n}{2} - 1$ comparisons).

The algorithm totally makes $\frac{3}{2}n - 2$ comparisons.

Here's the code implemented for the algorithm.

```python
from random import shuffle

count = 0


def lessthan(a, b):
    global count
    count += 1
    return a < b


def greaterthan(a, b):
    global count
    count += 1
    return a > b


def find_min_and_max(array):
    n = len(array)
    mi, mx = None, None
    for i in range(0, n, 2):
        if not lessthan(array[i], array[i + 1]):
            # swap the pairs
            array[i], array[i + 1] = array[i + 1], array[i]
    for i in range(0, n, 2):
        if mi is None or lessthan(array[i], mi):
            mi = array[i]
        if mx is None or greaterthan(array[i + 1], mi):
            mx = array[i + 1]
    return mi, mx


n = 20
```

```
array = [i for i in range(n)]
shuffle(array)

print(f"array: {array}")

print(f"min and max: {find_min_and_max(array)}")
print(f"3 n / 2 - 2 = {3 * n // 2 - 2}")
print(f"number of comparisons: {count}")
```

```
array: [17, 5, 10, 2, 11, 13, 16, 7, 14, 18, 12, 3, 1, 6, 4, 15, 0, 9, 8, 19]
min and max: (0, 19)
3 n / 2 - 2 = 28
number of comparisons: 28
```

Exercise 2

solution

Exercise 3

solution

Exercise 4

Exercise 5

Exercise 6

Exercise 7