

Question 1

Solution to (a)



Figure 1: Solution (a)

Solution to (b)

Denote Disk 1-4 (ascending in size) as D1, D2, D3, D4. and Tower 1-3 as T1, T2, T3. And denote the set of disk on one tower as D12 (disk 1 and 2), D123 (disk 1, 2, 3), etc.

To win the game, we need move all disk from T1 to T3, and since the placement of disks are strictly in descending order, the largest disk D4 should be the first disk to move to t3. And the second largest follows, then the third largest, and the smallest disk D1 should be the last disk to move to T3.

Therefore, the total number of steps to move all 4 disks is:

$$T(4) = T(\text{D4 T1 to T3}) + T(\text{D3 T1 to T3}) + T(\text{D2 T1 to T3}) + T(\text{D1 T1 to T3})$$

However, it is hard to track the number of steps for each disk to move to t3 Directly. Therefore, we can break down the problem into smaller subproblems by considering the the movement of D123 from T1 to T3 as a whole, this gives us the following equation:

$$T(4) = T(\text{D4 T1 to T3}) + T(\text{D123 T1 to T3})$$

And for D4 to move to T3, we also need to move D1-3 to T2 first, and then move D4 to T3, and then move D1-3 to T3. Therefore, the total number of steps for D4 to move to T3 is:

$$T(4) = 1 + T(\text{D123 T1 to T2}) + T(\text{D123 to T3})$$

Considering T123 from T1 to T2 and T1 to T3, because there's no T4 restrict the movement, the number of steps for T123 to move to T2 and T3 are the same. Therefore, we have:

$$\begin{aligned} T(4) &= 1 + T(\text{D123 T1 to T3}) + T(\text{D123 to T3}) \\ &= 1 + 2T(3) \\ &= 15 \end{aligned}$$

And because $T(3)$ is given as the minimum number of steps to move 3 disks from T1 to T3, we have $T(3) = 7$. Therefore, the minimum number of steps to move 4 disks from T1 to T3 is 15.

Solution to (c)

To find the recursive relation, we should first find the recursive relation for $T(3)$. It is easy to see that to move D3 to T3, we need to move D12 to T2 first, then move D3 to T3, and then move D12 to T3. This gives us the following relation:

$$T(3) = 1 + 2T(2)$$

And the recursive relation for $T(2)$ is almost trivial, we simply move D1 to T2, then move D2 to T3, and then move D1 to T3. Therefore, we have:

$$T(2) = 1 + 2T(1)$$

Based on the solution to (b), we can derive the following relation:

$$T(4) = 1 + 2T(3)$$

$$T(3) = 1 + 2T(2)$$

$$T(2) = 1 + 2T(1)$$

$$T(1) = 1$$

Then consider we have k where $k > 4, k \in \mathbb{Z}^+$ disk. In order to move D_k to T3, we need to move D1 to D_k to T2 first, then move D_k to T3, and then move D1 to D_k to T3. This is the minimal move for every level of k , since you must have T3 cleared as well as every disk above D_k removed before move D_k to T3, which results in a D1 to D_k on T2. In addition, move D1 to D_k to D2 and to D3 are 2 same sets of movements since there's no D_k to restrict the movement of D1 to D_k and one can switch the role of T2 and T3.

Therefore we propose a recurrence relation for $T(n)$:

$$T(n) = 1 + 2T(n-1)$$

Then we can solve this recurrence relation by substitution method:

$$\begin{aligned} T(n) &= 1 + 2T(n-1) \\ &= 1 + 2(1 + 2T(n-2)) = 1 + 2 + 4T(n-2) \\ &= 1 + 2 + 4(1 + 2T(n-3)) = 1 + 2 + 4 + 8T(n-3) \\ &= \dots \\ &= (1 + 2 + 4 + 8 + \dots + 2^{k-1}) + 2^k T(n-k) \end{aligned}$$

where $0 < k < n, k \in \mathbb{Z}$

The recursion stops when $n - k = 1 \Rightarrow k = n - 1$. Therefore, the minimal total number of steps to move n disks from T1 to T3 is:

$$\begin{aligned} T(n) &= \sum_{i=0}^{n-2} 2^i + 2^{n-1} \cdot 1 \\ &= \frac{(2^{n-1}) - 1}{2 - 1} + 2^{n-1} \text{ geometric series sum of } n-1 \text{ terms} \\ &= 2^{n-1} - 1 + 2^{n-1} \\ &= 2^n - 1 \end{aligned}$$

Solution to (d)

Let $n = \log(m)$. The relation becomes

$$T(\log(m)) = 2T(\log(m) - 1) + 1 = 2T\left(\log\left(\frac{m}{2}\right)\right) + 1$$

Define $S(m) = T(\log(m))$, then we have

$$S(m) = 2S\left(\frac{m}{2}\right) + 1$$

This has the form required by master theorem, where $a = 2, b = 2, f(n) = 1$ and is asymptotic positive. Check case of 1 master theorem: If $f(n) = O(n^{\log_b a - \epsilon})$ for some $\epsilon > 0$, then $S(m) = \Theta(n^{\log_b a})$. - This is true because $f(n) = O(n^{\log_b a - \epsilon})$ for $\epsilon = 1$, since $1 = O(n^{1-1})$.

Therefore, the solution is:

$$S(m) = \Theta(m)$$

since $n = \log(m) \rightarrow m = 2^n$, we have:

$$T(n) = S(2^n) = \Theta(2^n)$$

So from (c) we have $T(n) = 2^n - 1$, and by the definition of Θ , we can find $c_1 > 0, c_2 > 0, n_0 \geq 0$ such that:

$$c_1 \cdot 2^n \leq 2^n - 1 \leq c_2 \cdot 2^n \quad \text{for all } n \geq n_0$$

let's choose $c_1 = 0.5, c_2 = 1, n_0 = 1$, then we have:

$$2^{n-1} \leq 2^n - 1 \leq 2^n \quad \text{for all } n \geq 1$$

The upper bound is trivial to verify, and the lower bound is also true since:

$$\begin{aligned} c_0 \cdot 2^n &\leq 2^n - 1 \\ 1 &\leq 2^n(1 - c_0) \\ 1 - c_0 &\geq \frac{1}{2^n} \\ c_0 &\leq 1 - \frac{1}{2^n} \end{aligned}$$

sub in $n = 1$, we have $c_0 \leq 0.5$, therefore, the $c_0 = 0.5$ works

Question 2

Solution to (a)

1. Pivot $\text{arr}[6]$: 4
 - [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[0] = 3$ and $\text{arr}[6] = 4$
 - Since $\text{arr}[0] = 3 < \text{arr}[6] = 4$, swap with self
 - [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[1] = 1$ and $\text{arr}[6] = 4$
 - Since $\text{arr}[1] = 1 < \text{arr}[6] = 4$, swap with self
 - [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[2] = 5$ and $\text{arr}[6] = 4$
 - No swap
 - [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[2] = 5$ and $\text{arr}[3] = 7$
 - No swap
 - [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[2] = 5$ and $\text{arr}[4] = 6$
 - No swap
 - [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[2] = 5$ and $\text{arr}[5] = 2$
 - Since $\text{arr}[2] = 5 > \text{arr}[5] = 2$, swap
 - Last swap to put pivot in place: $\text{arr}[3] = 7$ and $\text{arr}[6] = 4$
2. Pivot $\text{arr}[2]$: 2
 - [3, 1, 2]
 - Compare $\text{arr}[0] = 3$ and $\text{arr}[2] = 2$
 - No swap
 - [3, 1, 2]
 - Compare $\text{arr}[0] = 3$ and $\text{arr}[1] = 1$
 - Since $\text{arr}[0] = 3 > \text{arr}[1] = 1$, swap
 - Last swap to put pivot in place: $\text{arr}[1] = 3$ and $\text{arr}[2] = 2$
3. Pivot $\text{arr}[6]$: 7
 - [6, 5, 7]
 - Compare $\text{arr}[4] = 6$ and $\text{arr}[6] = 7$
 - Since $\text{arr}[4] = 6 < \text{arr}[6] = 7$, swap with self
 - [6, 5, 7]
 - Compare $\text{arr}[5] = 5$ and $\text{arr}[6] = 7$
 - Since $\text{arr}[5] = 5 < \text{arr}[6] = 7$, swap with self
 - Last swap to put pivot in place: $\text{arr}[6] = 7$ and $\text{arr}[6] = 7$
4. Pivot $\text{arr}[5]$: 5
 - [6, 5]
 - Compare $\text{arr}[4] = 6$ and $\text{arr}[5] = 5$
 - No swap
 - Last swap to put pivot in place: $\text{arr}[4] = 6$ and $\text{arr}[5] = 5$

Since the sort is done in each partition, the final sorted array is: $A = [1, 2, 3, 4, 5, 6, 7]$

Solution to (b)

This is a sorted array, so the partition will always be on the left side of the pivot. Therefore, the number of comparisons is the sum of the first $n-1$ integers

$A_0 = [1, 2, 3, 4, 5, 6, 7]$ number of comparison is 6

$A_1 = [1, 2, 3, 4, 5, 6]$ number of comparison is 5

$A_2 = [1, 2, 3, 4, 5]$ number of comparison is 4

$A_3 = [1, 2, 3, 4]$ number of comparison is 3

$A_4 = [1, 2, 3]$ number of comparison is 2

$A_5 = [1, 2]$ number of comparison is 1

$$n = 1 + 2 + 3 + 4 + 5 + 6 = 21$$

Solution to (c)

Suppose the partition is based on a value p , where $0 < p < 1$, so approximately each time partition will have np and $n(1-p)$ elements. We can then express the number of comparison as:

$$T(n) = T(pn) + T((1-p)n) + n - 1$$

And we know that quick sort has the best case $O(n \log n)$. We then try substituting $T(n) = n \log n$, since we want to know the best case with least comparison, which gives us:

$$T(n) = (pn) \log pn + (1-p)n \log(1-p)n + n - 1$$

To find which p gives the minimum number of comparison, we can take the derivative of $T(n)$ with respect to p and set it to 0:

$$\frac{\partial}{\partial p} T(n) = 0$$

$$n^2 \log p + n^2 \log(1-p) = 0$$

$$p = 0.5$$

Therefore, the best case is to have the partition as close to the middle as possible, which is to have the pivot as the median of the array.

Therefore, given a sorted array,

$$[a_1, a_2, a_3, a_4, a_5, a_6, a_7]$$

where $a_i < a_j$ for $i < j$, the best case is to have the pivot as a_4 , which is the median of the array. We will put a_4 as the pivot, then we will have two partitions which we don't know the order,

$$A = [[a', a'', a'''], a_4, [a''''', a''''', a''''']]$$

Similarly, we will put the median of the two partitions as the pivot,

$$A_1 = [[a', a''], a_2], A_2 = [[a''', a'''''], a_6]$$

Since there's only one way to partition a size of 2 array, the order can be arbitrary

And then we can merge the two partitions to get the original array. Here a_6, a_4 are swapped to get the original array

$$A = [a', a'', a_2, a_6, a''', a''''', a_4]$$

This will give us the minimum number of comparisons 10. A permutation run verify that this is the best case.

Solution to (d)

As showed in part (c), the best case is to have the pivot as the median of the array, which is the middle element of the array. Therefore, the recurrence relation can be expressed and solved as follows:

$$\begin{aligned}
 T(n) &= n - 1 + 2T\left(\frac{n}{2}\right) \\
 &= n - 1 + 2\left(\frac{n}{2} - 1 + 2T\left(\frac{n}{4}\right)\right) = (n - 1) + (n - 2) + 2T\left(\frac{n}{4}\right) && \text{by substitution} \\
 &= n - 1 + n - 2 + n - 3 + 2T\left(\frac{n}{8}\right) = (n - 1) + (n - 2) + (n - 3) + 2T\left(\frac{n}{8}\right) \\
 &= (n - 1) + (n - 2) + (n - 3) + \dots + (n - k) + 2T\left(\frac{n}{2^k}\right)
 \end{aligned}$$

The recursion stops when $\frac{n}{2^k} = 1 \Rightarrow k = \log n = \log(2^k - 1)$.

Considering $n = 1$, we have $T(1) = 1$ as one comparison is needed to check the size of the array. Thus

$$\begin{aligned}
 T(n) &= \sum_{i=1}^{\log n} (n - i) + 2 \\
 &= \sum_{i=1}^{\log n} n - \sum_{i=1}^{\log n} i + 2 \\
 &= n \log n - \log n \frac{\log n + 1}{2} + 2 \\
 &= (2^k - 1) \log(2^k - 1) - \log(2^k - 1) \frac{\log(2^k - 1) + 1}{2} + 2
 \end{aligned}$$

It's easy to see that $T(n) = \Theta(n \log n)$

In the worst case, the pivot is the smallest or largest element of the array, which is the first or last element of the array. Therefore, the recurrence relation is:

$$T(n) = n - 1 + T(n - 1) = n + (n - 1) + \dots + 1 = \sum_{i=1}^{n-1} i$$

This is a arithmetic series, and the sum of the series is $T(n) = \frac{n(n-1)}{2} = \frac{(2^k-1)(2^k-2)}{2}$. Therefore, the worst case is $T(n) = O(n^2)$

Question 3

Solution to (a)

1. [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[0] = 3$ and $\text{arr}[6] = 4$
 - No swap
2. [3, 1, 5, 7, 6, 2, 4]
 - Compare $\text{arr}[0] = 3$ and $\text{arr}[5] = 2$
 - Since $\text{arr}[0] = 3 > \text{arr}[5] = 2$, swap
3. [2, 1, 5, 7, 6, 3, 4]
 - Compare $\text{arr}[5] = 3$ and $\text{arr}[1] = 1$
 - No swap
4. [2, 1, 5, 7, 6, 3, 4]
 - Compare $\text{arr}[5] = 3$ and $\text{arr}[2] = 5$
 - Since $\text{arr}[5] = 3 > \text{arr}[2] = 5$, swap
5. [2, 1, 3, 7, 6, 5, 4]
 - Compare $\text{arr}[2] = 3$ and $\text{arr}[4] = 6$
 - No swap
6. [2, 1, 3, 7, 6, 5, 4]
 - Compare $\text{arr}[2] = 3$ and $\text{arr}[3] = 7$
 - No swap

Solution to (b)

1. PARTITION(1,7) = [3, 1, 5, 7, 6, 2, 4] Compare 6 times
2. PARTITION(1,2) = [2, 1] Compare 1 times
3. PARTITION(4,7) = [7, 6, 5, 4] Compare 3 times
4. PARTITION(4,6) = [4, 6, 5] Compare 2 times
5. PARTITION(5,6) = [6, 5] Compare 1 times

Therefore, the total number of comparisons is $6 + 1 + 3 + 2 + 1 = 13$

Solution to (c)

```
def PARTITION(arr, p, r):
    direction = -1
    print(f" + PARTITION({p}:{r+1}) = ", arr[p:r+1])
    count = 0
    while p != r:
        count += 1
        if (arr[p] > arr[r] and p < r) or (arr[p] < arr[r] and p > r):
            print(f" - swap {arr[p]} and {arr[r]}")
            arr[p], arr[r] = arr[r], arr[p]
            p, r = r, p
            direction *= -1
        r += direction
    print(f" + After compare {count} times, A = ", arr)
    return p

def HUNGARIAN_QUICKSORT(arr, p, r):
    if p < r:
        q = PARTITION(arr, p, r)
        HUNGARIAN_QUICKSORT(arr, p, q - 1)
        HUNGARIAN_QUICKSORT(arr, q + 1, r)
    pass

arr = [3, 1, 5, 7, 6, 2, 4]
HUNGARIAN_QUICKSORT(arr, 0, len(arr) - 1)
```

Output:

```
+ PARTITION(0:7) = [3, 1, 5, 7, 6, 2, 4]
- swap 3 and 2
- swap 3 and 5
+ After compare 6 times, A = [2, 1, 3, 7, 6, 5, 4]
+ PARTITION(0:2) = [2, 1]
- swap 2 and 1
+ After compare 1 times, A = [1, 2, 3, 7, 6, 5, 4]
+ PARTITION(3:7) = [7, 6, 5, 4]
- swap 7 and 4
+ After compare 3 times, A = [1, 2, 3, 4, 6, 5, 7]
+ PARTITION(3:6) = [4, 6, 5]
+ After compare 2 times, A = [1, 2, 3, 4, 6, 5, 7]
+ PARTITION(4:6) = [6, 5]
- swap 6 and 5
+ After compare 1 times, A = [1, 2, 3, 4, 5, 6, 7]
```

Question 4

Solution to (a)