Problem 0

B11901164 陳秉緯

Problem 1

1.1 no collaborator

1.2 collaborators: 林育正

1.3 no collaborator

1.4 no collaborator

1.5 no collaborator

1.6 no collaborator

Problem 2

2.1 collaborators: 謝沅瓏

2.2 collaborators: 林育正

2.3 no collaborator

2.4 collaborators: 謝沅瓏, 林育正

Problem 3

Collaborators: 謝沅瓏

Problem 4

Collaborators: 謝沅瓏, 林育正, 林順文, 王品翔

Problem 1

1.

There are $\sqrt{n}$ iterations, $\sqrt{n}\times\sqrt{n} \geq n$ and return $\sqrt{n}$

Hence the time complexity is $\Theta(\sqrt{n})$

---

2. $\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=i}^{j} 1$

$= \sum_{i=1}^{n}\sum_{j=i}^{n}(j-i)$

$= \sum_{i=1}^{n}\left[\frac{(i+n)(n-i+1)}{2} - (n-i+1)i\right]$

$= \sum_{i=1}^{n}\left[\frac{1}{2}(in-i^2+i+n^2-in+n) - ni + i^2 - i\right]$

$= \sum_{i=1}^{n}\left[\frac{1}{2}(-i^2+i+n+n^2) + i^2 - ni - i\right]$

$= \sum_{i=1}^{n}\left[\frac{1}{2}i^2 + (-n-\frac{1}{2})i + \frac{1}{2}(n+n^2)\right]$

$= \frac{1}{2}\frac{n(n+1)(2n+1)}{6} - (n+\frac{1}{2})\frac{(1+n)n}{2} + n\times\frac{1}{2}(n+n^2)$

$= n(n+1)\left[\frac{1}{12}(2n+1) - \frac{1}{2}(n+\frac{1}{2}) + \frac{1}{2}n\right]$

$= n(n+1)\left[\frac{1}{6}n + \frac{1}{12} - \frac{1}{4}\right]$

$= n(n+1)(\frac{1}{6}n - \frac{1}{6})$

$= \frac{1}{6}n(n+1)(n-1)$

$= \frac{1}{6}(n^3-n)$

$= n^3(\frac{1}{6} - n^{-2})$

We see that $\lim_{n\to\infty}\frac{\frac{1}{6}(n^3-n)}{n^3} = \frac{1}{6} > 0$

Using the proof in class that justifies how the limit def. leads to the formal def.

$\therefore \frac{1}{6}(n^3-n) = \underline{\Theta(n^3)}$

---

3. Let the total time cost of DOUBLE-TWODOWN(n) is $T(n)$

$T(n) = 2T(n-2) + T(0)$

$= 2(2T(n-4)) + T(0)$

$= 2^2T(n-4) + T(0)$

$= 2^3T(n-6) + T(0)$

$= (2^{\frac{n}{2}} + 1)T(0)$

$= 2^{\frac{n}{2}}(1 + 2^{-\frac{n}{2}})T(0)$

We see that

$\lim_{n\to\infty}\frac{(2^{\frac{n}{2}}+1)T(0)}{2^{\frac{n}{2}}} = T(0) > 0$ const.

$\underline{\therefore T(n) \in \Theta(2^{\frac{n}{2}})}$ by proof in class

4.

Def : $f(n) = O(g(n))$ if $\exists\, c > 0$ and $n_0 > 0$ s.t. $0 \le f(n) \le cg(n) \;\; \forall n \ge n_0$

$$\Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = [0, \infty), \text{ if the limit exists.}$$

$n! = O(2^n)$

$$\lim_{n\to\infty} \frac{n!}{2^n} = \lim_{n\to\infty} \frac{1\cdot 2\cdot 3\cdot 4 \cdots n}{2\cdot 2\cdot 2\cdot 2 \cdots 2}$$

$$= \lim_{n\to\infty} \frac{3}{2} \times \frac{5\cdot 6\cdot 7 \cdots n}{2\cdot 2\cdot 2 \cdots n}$$

$$= \infty$$

$\therefore\; n! \ne O(2^n)$ Q.E.D.

---

5. $\forall p \in \mathbb{N}, q > 1, n^p = O(q^n)$

$$\begin{cases} \ln(n^p) = p\ln(n) \\ \ln(q^n) = n\ln(q) \end{cases}$$

$\exists\, C = \frac{p}{\ln(q)}$ and $n_0 = 1$

s.t. $0 \le p\ln(n) \le c(\ln q)n$ for $n \ge n_0$

(with the fact that $\ln \le n-1 < n$ for $n \ge 1$)

$\Rightarrow\; e^0 \le e^{p\ln(n)} \le e^{c(\ln q)n}$ ($y = e^x$ is strictly increasing)

$\Rightarrow\; 1 \le n^p \le e^c \cdot q^n$

$\Rightarrow\; 0 \le n^p \le A q^n, \; A = e^c$

$\therefore\; n^p = O(q^n)$ Q.E.D.

---

6. $f_k(n) = O(n) \;\; \forall k \in \mathbb{N}$

$\exists\, c > 0$ and $n_0 > 0$
s.t. $0 \le f_k(n) \le cn \;\; \forall n \ge n_0$

Let $f_k(n) = cn$

$\sum_{k=1}^{n} f_k(n) = \sum_{k=1}^{n} cn = cn^2$

$\exists\, c' = c$ and $n_0' = n_0$

s.t. $0 \le cn^2 \le c'n \;\; \forall n \ge n_0'$

$\Rightarrow cn^2 = O(n^2)$

$\Rightarrow \sum_{k=1}^{n} f_k(n) = O(n^2)$ Q.E.D.

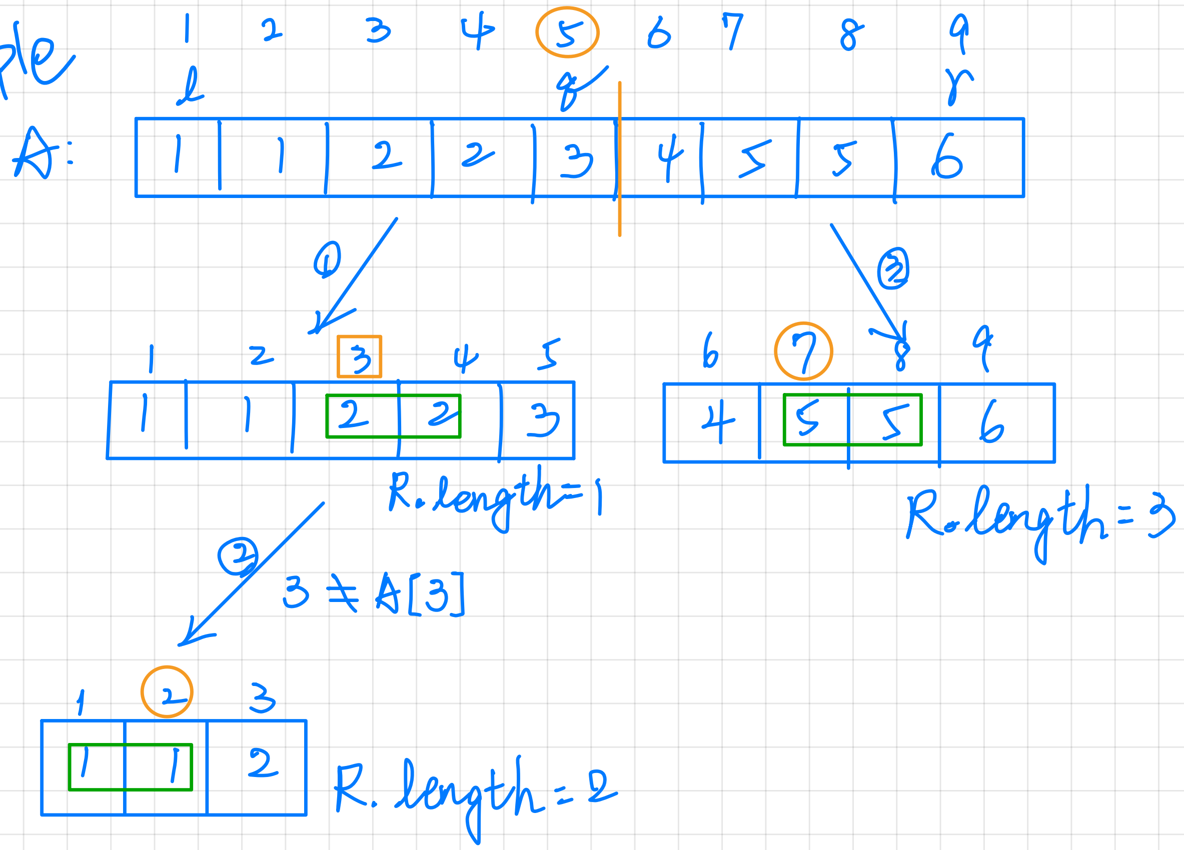
Problem 2

1. n, $n \geq 6$, $1 \sim n-3$

$$\begin{pmatrix} l = 1 \\ r = A.length \\ R \text{ is empty initially} \end{pmatrix}$$

```
FIND-THREE-REPEATED-NUMBERS(A, l, r, R)
  q = ⌊(l+r)/2⌋
  if ((q > 1) and (A[q] = A[q-1] or A[q] = A[q+1]))
      R.push(A[q])
      if R.length == 3
          return R
  if A[q] ≠ q
      FIND-THREE-REPEATED-NUMBERS(A, l, q, R)
  if R.length ≠ 3
      FIND-THREE-REPEATED-NUMBERS(A, q+1, r, R)
  else
      return R
```

example



find left subarray first if $A[q] \neq q$,
store $A[q]$ into R if $A[q] = A[q-1] \vee A[q] = A[q+1]$
if length of R is not equal to 3, then
keep finding reapted numbers in right subarray

2. FIND-REPEATED-NUMBER(A)

	cost	time
value_sum = 0; index_sum = 0;	c_1	1
for i = 1 to A.length	c_2	m
value_sum ^= A[i]	c_3	$m-1$
if i != A.length	c_4	$m-1$
index_sum ^= i	c_5	$m-2$
return value_sum ^ index_sum	c_6	1

$\therefore T(m) = c_1 + c_2 m + c_3(m-1) + c_4(m-1) + c_5(m-2) + c_6$

$= Am + B, \quad A, B = \text{const.}$

$\lim_{m \to \infty} \frac{T(m)}{m} = A \Rightarrow T(m) \in O(m)$

$\because$ the numbers of variables declared have nothing to do with m

$\therefore$ $O(1)$ extra space

---

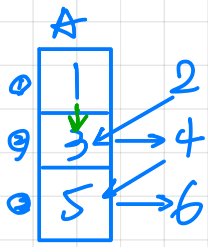
3. Game(L, k)

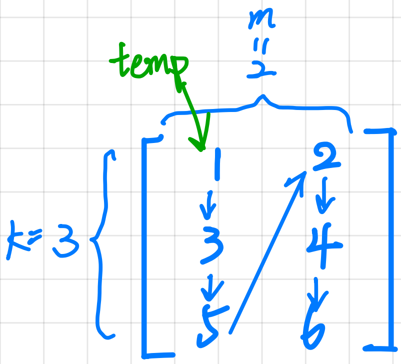
```
for i = k downto 0
        L->head->prev = L->tail
        L->tail->next = L->head
        L->tail = L->tail->prev
        L->tail->next->prev = NULL
        L->tail->next = NULL
        L->head = L->head->prev
```

There are k iterations, $T(n) \in O(k)$ and $O(1)$ extra space per shouting

4

```
TRANSPOSING(L, k, m)
    A is an array of pointer whose length = k
    temp = L.head
    for j = 1 to m
        for i = 1 to k
            A[i] = temp
            if i > 1 and i < k
                A[i-1].next = A[i]
            if i = k
                A[i] = temp.next
        temp = temp.next
    return L
```





k iterations for m times

⇒ time complexity $O(k \times m) = O(n)$

new an array whose length = k

⇒ $O(k)$ extra space