Data Structures HW #1

① ⓐⓑ

int max_id	
For i=n-1 to 1 by -1	n
max_id = i	n-1
for j=i-1 to 0 by -1	$\left(\frac{n(n-1)}{2}\right) + (n-1)$
if (A[max_id] < A[j]	$\left(\frac{n(n-1)}{2}\right)$
max_id = j	$\left(\frac{n(n-1)}{2}\right)$
swaps indexes	n-1

because everytime the outer loop runs, it checks the inner loop that extra amount of times.

$$n + 2(n-1) + 3\left(\frac{n(n-1)}{2}\right)$$

$$n+3n-3+3\left(\frac{n^2-n}{2}\right) = 4n-3+\frac{3n^2-3n}{2} = 8n-6+3n^2-3n =$$

$$3n^2+5n-6$$

same in $2(i;;)$ $\sum_{i=1}^{\log_2 n}(2i) = \log_2 n$

ⓒ best case: $T(n) = O(n^2)$
worst case: $T(n) = O(n^2)$
} same doesn't change the amount of times it loops

② ⓘ

j=1	1	C_1
for i=2 to 10n by 2	$\frac{10n}{2}-2 = 5n-2$	C_2
j=j+i;	5n-2	C_3

$$1+2(5n-2) = 1+10n-4 = 10n-3 = \boxed{n}$$

ⓘⓘ

i=1	1	C_1
while (i<2^n)	$\log_3 2^n$	C_2
i=i*3	$\log_3 2^n - 1$	C_3

$$1+2(\log_3 2^n)-1 = 2\log_3 2^n = \boxed{\log_3 2^n}$$

ⓘⓘⓘ

i=2	1	C_1
k=2	1	C_2
while (i<n)	$\log_2 n$	C_3
For j=1 to i	$\sum_{i=1}^{\log_2 n-1}(2^i)+\log_2 n$	C_4
k=k-j	$\sum_{i=1}^{\log_2 n-1}(2^i)$	C_5
i=i*2	$\log_2 n - 1$	C_6

increasing exponentially $\log n$

n

$$1+2\log_2 n \cdot \left(\sum_{i=1}^{\log_2 n-1}(2^i)+\log_2 n\right)+$$

$$\sum_{i=1}^{\log_2 n-1}(2^i) = \boxed{n}$$