2.4

According to decision tree, the number of comparisons are

$\log_2 5! = \log_2 120 \qquad 2^6 = 64 < 120 < 2^7 = 128$

So we need at least 7 comparision

the number of comparition of Ford - Johnson Algorithms

is the sum of :

(1) $\lfloor \frac{n}{2} \rfloor$ comparition of items in each pair
(2) $\lfloor \frac{n}{2} \rfloor$ comparition among paris of item
(3) $C(\lfloor \frac{n}{2} \rfloor)$ comparition for the recursive call
(4) number of comparitions for binary insertions to insert remaining elements.

for (3) worse case of each inserctions is, since the first time

number of comparition is 0 $\qquad 2\ 2 \qquad 2\ 2\ 3,$

$C(n) = \sum_{i=1}^{n-1} \log_2 n+i = \log_2 \frac{(2n-1)!}{n!}$

for (4) if n is odd number of comparition is $\log_2 (n-1)$

So, when $n=5$ $\quad T(n) = \lfloor \frac{5}{2} \rfloor + \lfloor \frac{5}{2} \rfloor + \log_2 \frac{3!}{2!} + \log_2 4 = 7$

So Ford - Johnson Algorithm does achive this lower bound

2.5 : Assume the Largest number is the k^{th} number of the list $(1 \le k \le n)$

To find the lergest number of the list :

First we compare $k-1$ times, so up to now we find the largest number in list from position $k-1$

However, we are not sure whether there is a number larger than the k^{th} number in the List. So we again need to compare $n-k$ times to find the largest number.

Finally, total comparition is.

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