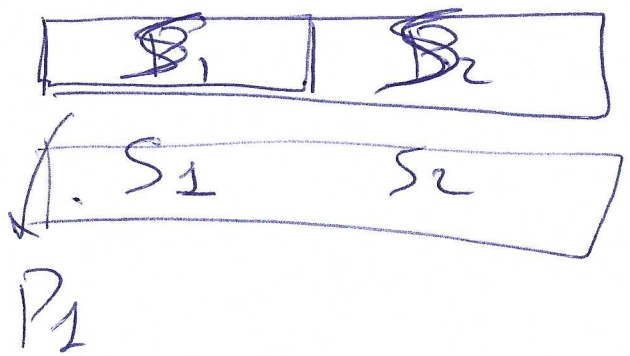
D&C

```
DC( P) {
  if ( P is small ) return lookup[P];
  (P1, ..., Pn) = split(P);          } think
  for i = 1..n
      Si = DC(Pi)                    } assume it works
  S = combine(S1, ..., Sn);          } think
  return S;
}
```

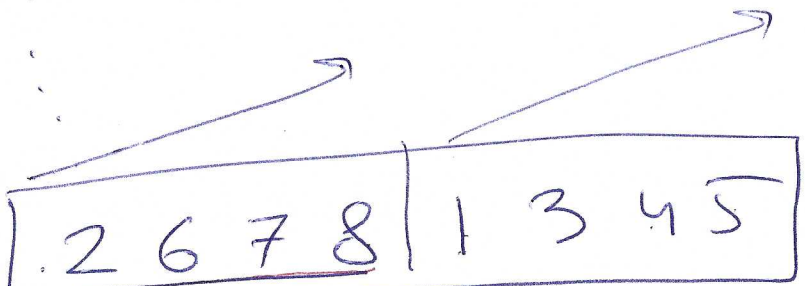


Sorting array

8 7 2 6 | 1 4 3 5 "trivial" split

2 6 7 8 | 1 3 4 5

Combine



shift

1 2 6 7 8 3 4 5	1	$\frac{n}{2}$ copies
1 2 3 6 7 8 4 5	3	$\frac{n}{2}$ copies
1 2 3 4 6 7 8 5	4	$\frac{n}{2}$ copies
1 2 3 4 5 6 7 8	5	$\frac{n}{2}$ copies

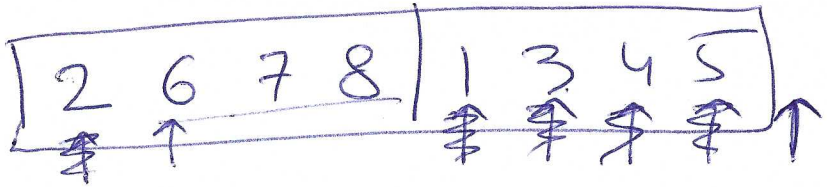
worst $\frac{n}{2}$ times

combine = $O(n^2)$

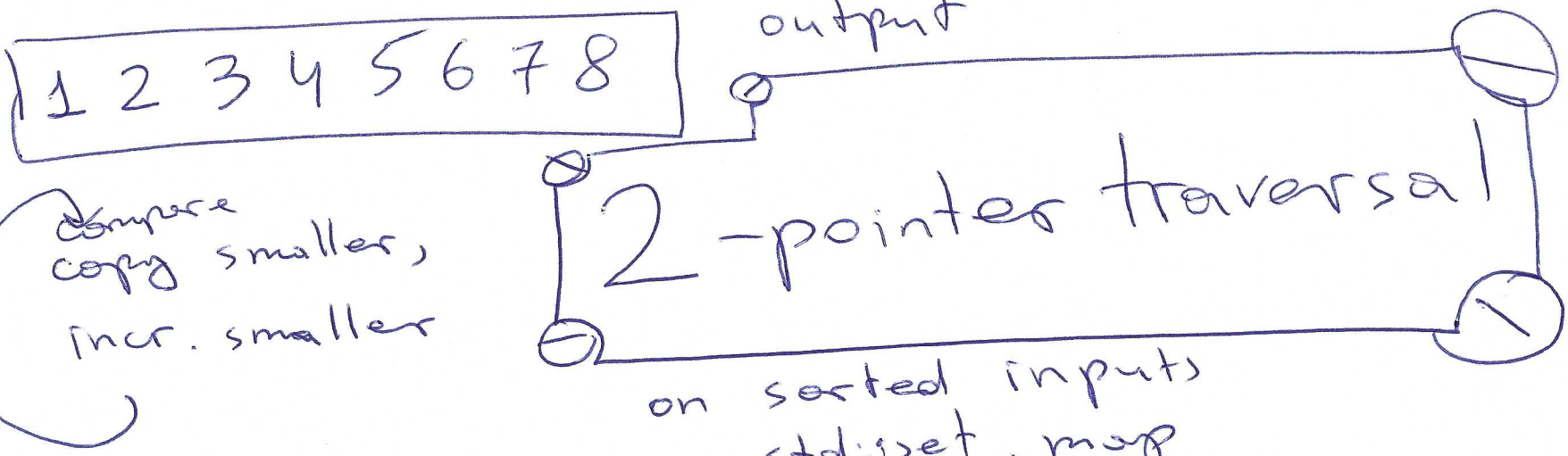
Recursive sort with "insertion sort" as combine

$T(n) = 2T(\frac{n}{2}) + n^2/4$ ... $T(n) = O(n^2)$

~~$O(n)$~~ $O(n \log n)$



extra memory is allowed



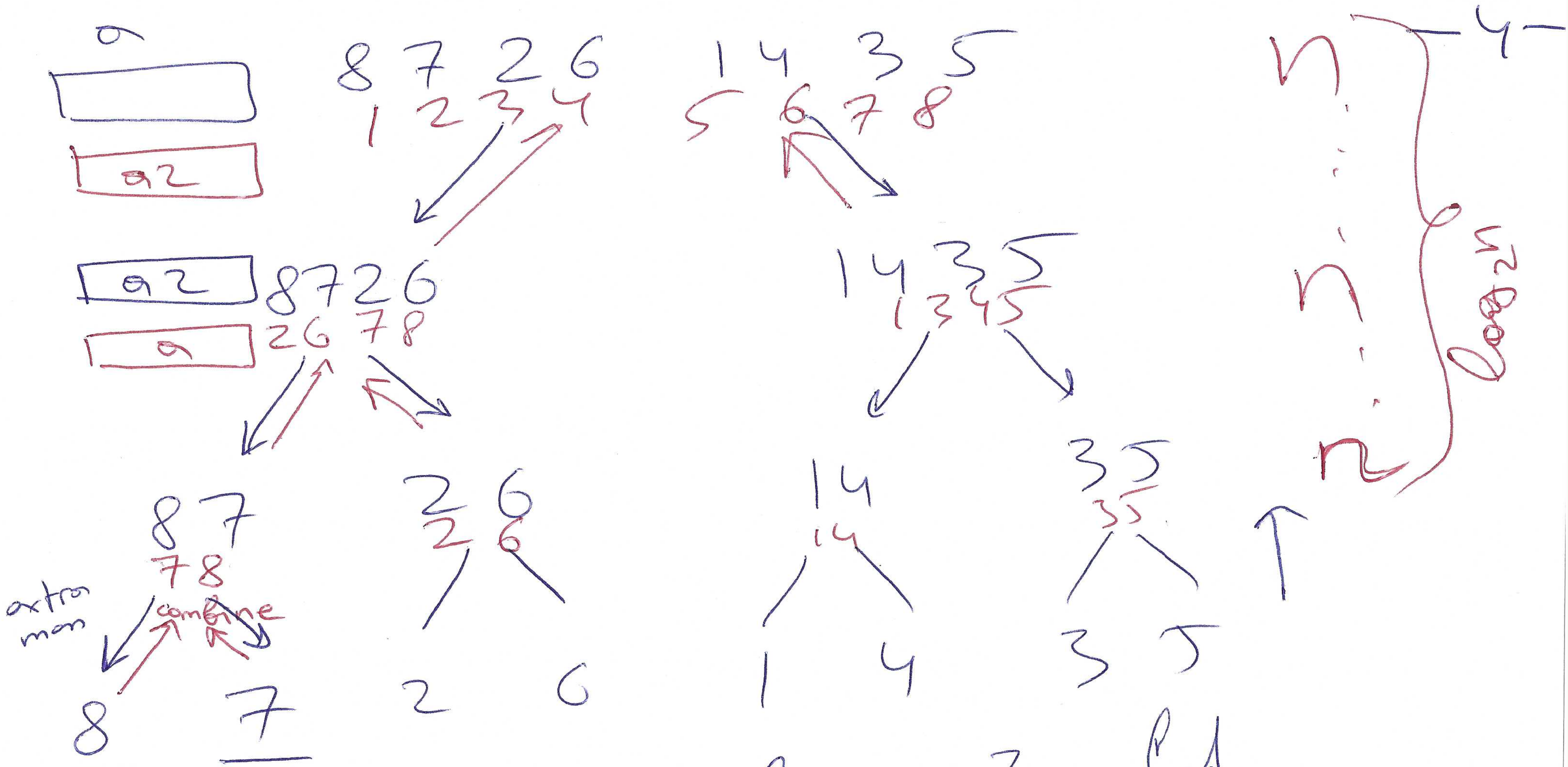
STL has "set union"
"set intersection"}
"set diff"

Mergesort = recursive sort with 2-ptr traversal as combine

$$\begin{cases} T(n) = 2T\left(\frac{n}{2}\right) + n \\ T(1) = 1 \end{cases}$$

← worst case, also average, also best

$T(n) = \underset{\text{worst case}}{O(n \log n)}$ Best possible, but extra memory



extra memory $n \log_2 n$ } a lot.

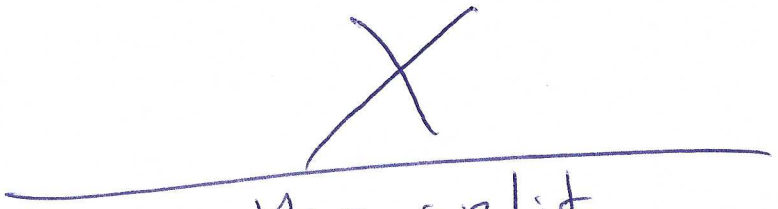
"alternating" n ⟸ still a lot.

combine "in place"

2 6 7 8 1 3 4 5

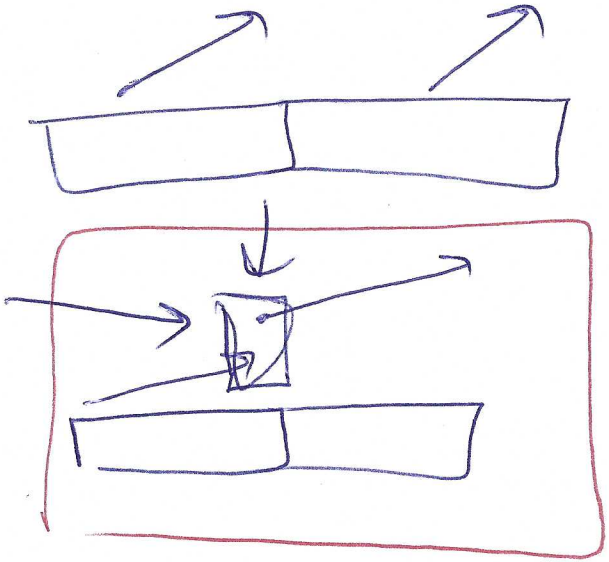
1 6 7 8 2 3 4 5

1 2 7 8 6 3 4 5



another split

combine "in place"



left.last < right.first

combine = NOP;

```
1   Partition( a, first, last ) { // index last belongs to array
2       i = first;
3       pivot = a[last];
4       for (j=first; j<last; ++j) {
5           if ( a[j]<pivot ) {
6               swap( a[j], a[i] );
7               ++i;
8           }
9       }
10      swap( a[i], a[last] );
11      return i;
12  }
13
```

O(n)

i ← pos for small P

8 7 2 6 1 4 3 5

2 7 8 6 1 4 3 5

2 1 8 6 7 4 3 5

2 1 4 6 7 8 3 5

2 1 4 3 7 8 6 5

index → 0 1 2 3 4

2 1 4 3 5 8 6 7

<5 >5

5 3 1 4 2 6 7

n-1

1 2 3 4 5 6 7

2 1 4 3 6 5 7

1 2 3 4 5 6 7

~~Partitio~~
Partition (our split for next sort)

T(n) QS (a, 0, n-1) { // 0 ... n-1
1 if (n==1) return ;
n q = Partition(a, 0, n-1);
T(q-1) QS (a, 0, q-1); // index q is excluded from sorting
T(n-q) QS (a, q+1, n-1);
// no combine
}

~~goog~~ google 10 most import. alg of 20th cent.

4 QS. (6 lines of code!!!)

Tony Hoare

RT of QS

$T(n) = T(q) + T(n-q) + n$, $T(1) = 1$

~~worst~~ $q = \frac{n}{2}$ $T(n) = 2T(\frac{n}{2}) + n$, $T(n) = O(n \log n)$

~~best~~

best

~~best~~ $q = 1$, $T(n) = T(1) + T(n-1) + n$

worst $q = n-1$

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

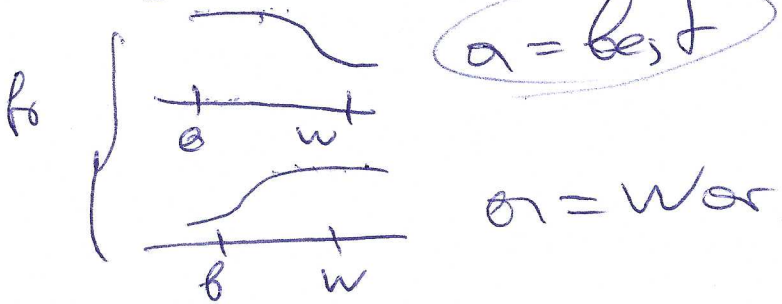
$= T(n-2) + n-1 + n =$

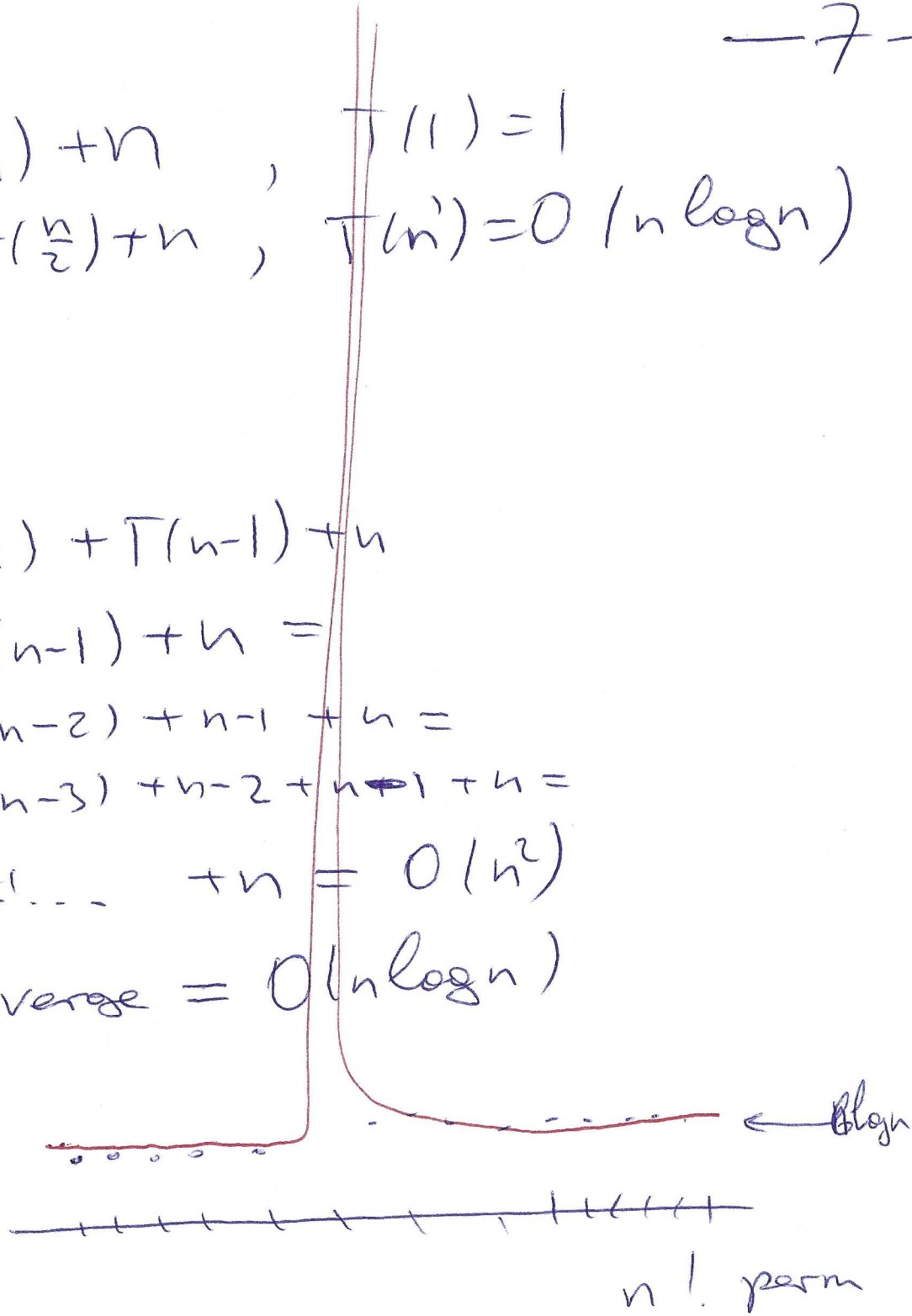
$= T(n-3) + n-2 + n-1 + n =$

$= 1 + 2 + \dots + n = O(n^2)$

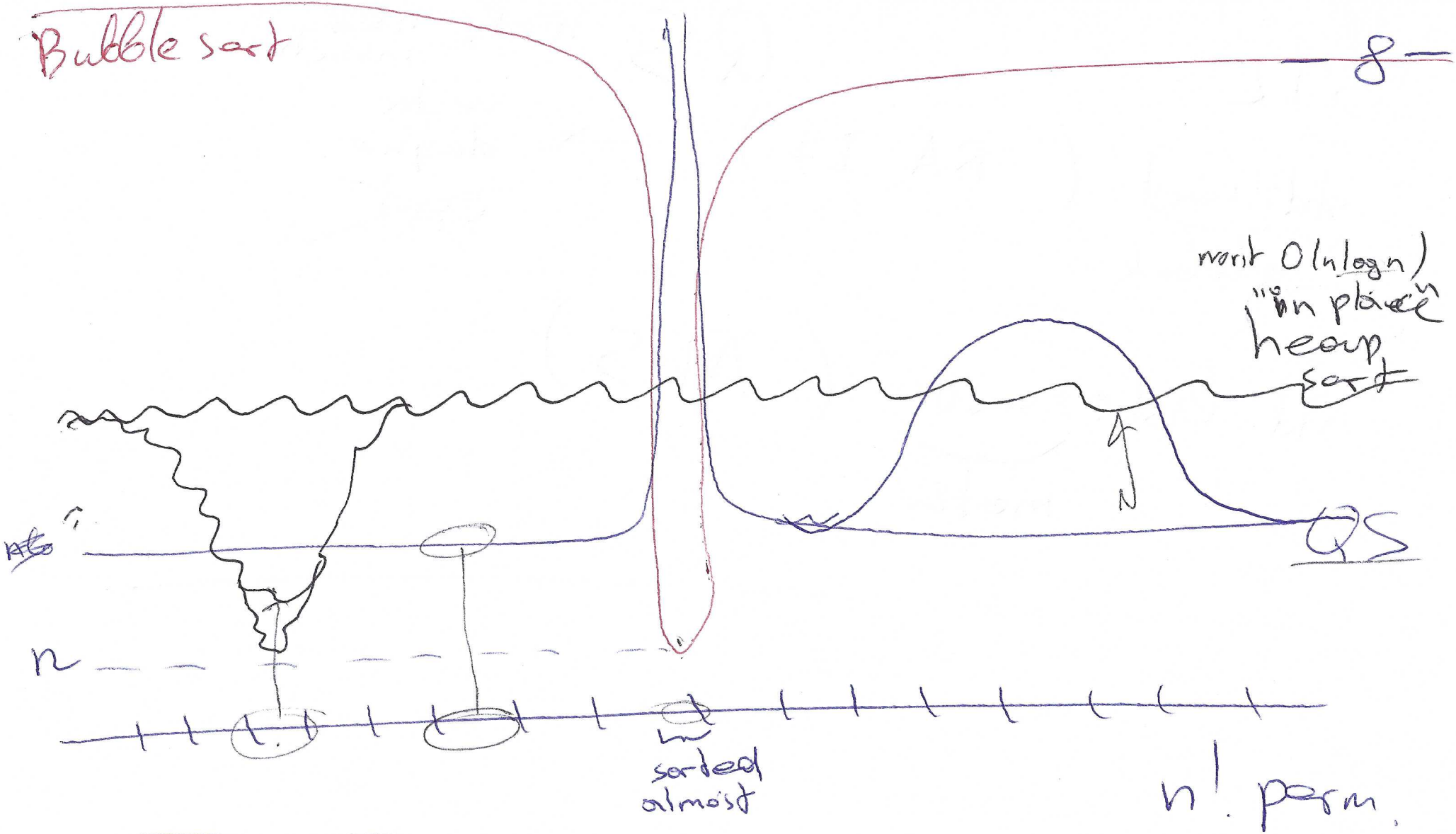
Average $= O(n \log n)$

average.





Bubble sort



Note: