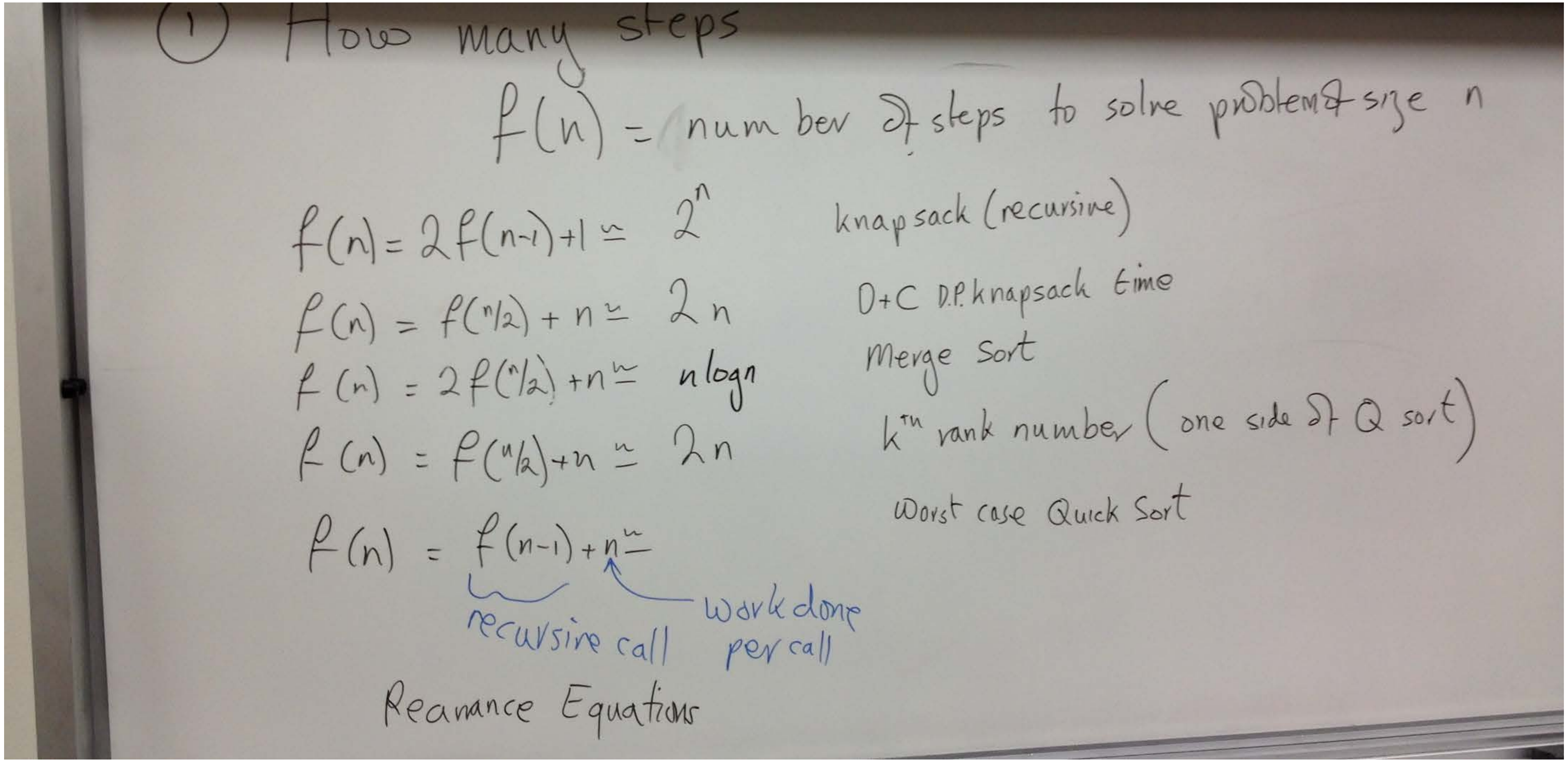
cs5050

02 06 14



① How many steps

$f(n)$ = number of steps to solve problem of size n

$f(n) = 2f(n-1)+1 \simeq 2^n$ knapsack (recursive)

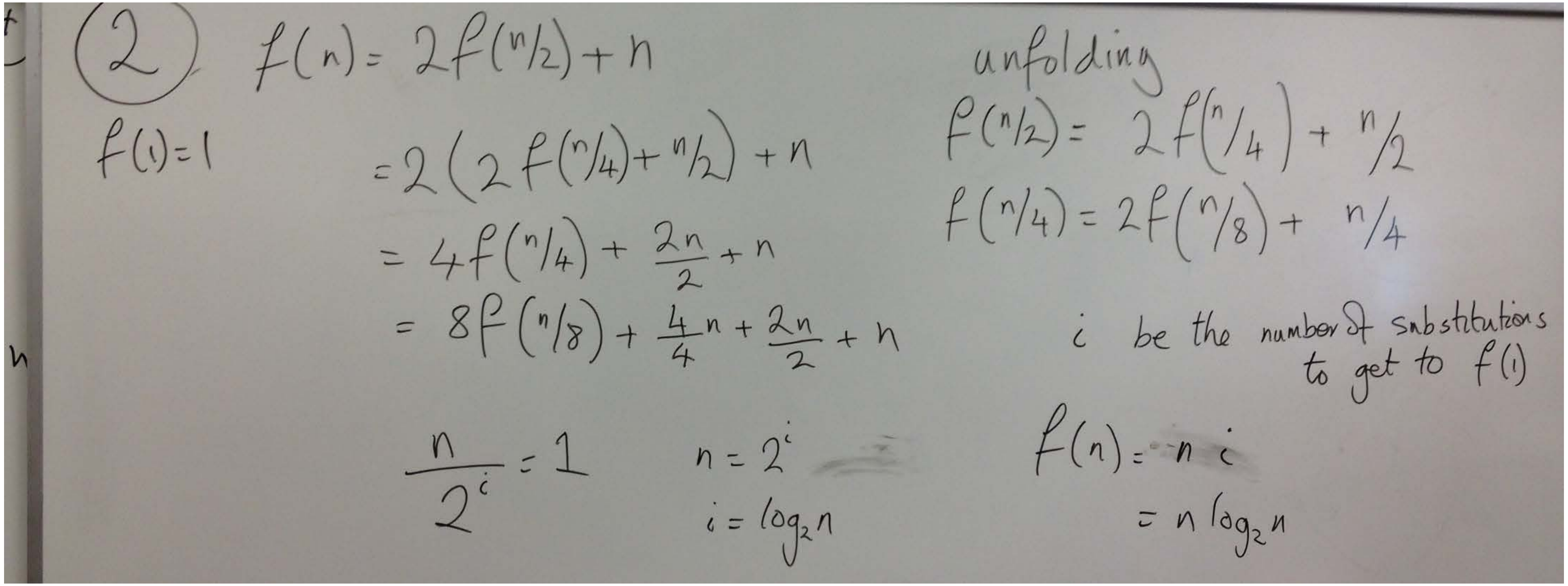
$f(n) = f(n/2) + n \simeq 2n$ D+C D.P. knapsack time

$f(n) = 2f(n/2)+n \simeq n\log n$ Merge Sort

$f(n) = f(n/2)+n \simeq 2n$ k^{th} rank number (one side of Q sort)

$f(n) = \underbrace{f(n-1)}_{\text{recursive call}} + \underset{\text{work done per call}}{n} \simeq$ Worst case Quick Sort

Recurance Equations



(2) $f(n) = 2f(n/2) + n$

$f(1) = 1$

$= 2(2f(n/4) + n/2) + n$

$= 4f(n/4) + \frac{2n}{2} + n$

$= 8f(n/8) + \frac{4}{4}n + \frac{2n}{2} + n$

unfolding

$f(n/2) = 2f(n/4) + n/2$

$f(n/4) = 2f(n/8) + n/4$

i be the number of substitutions to get to $f(1)$

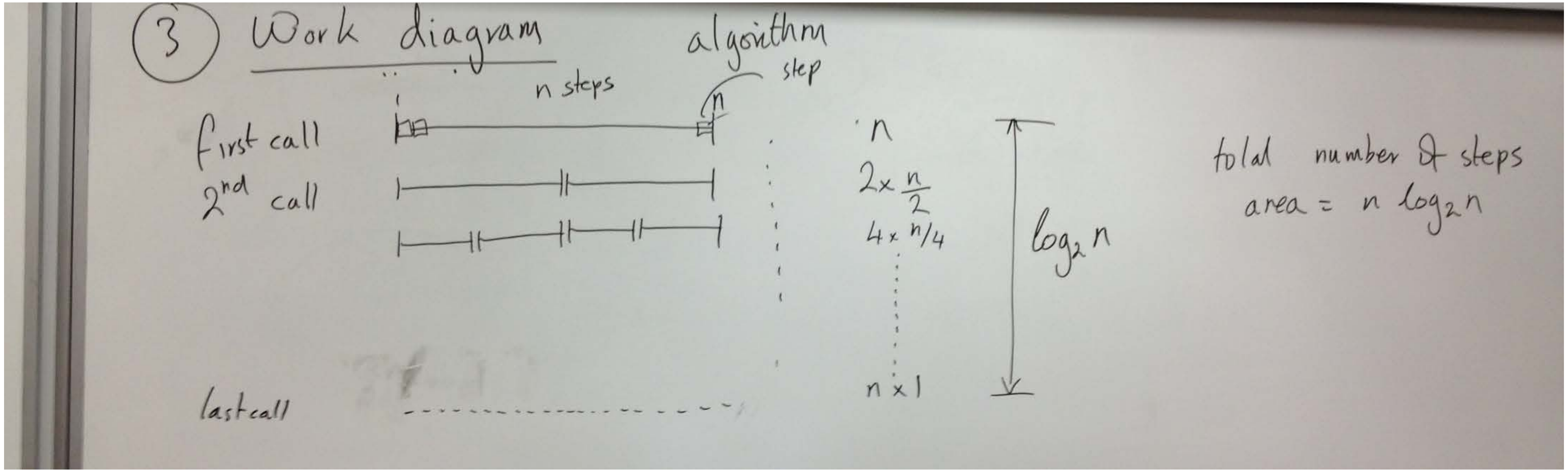
$\frac{n}{2^i} = 1$

$n = 2^i$

$i = \log_2 n$

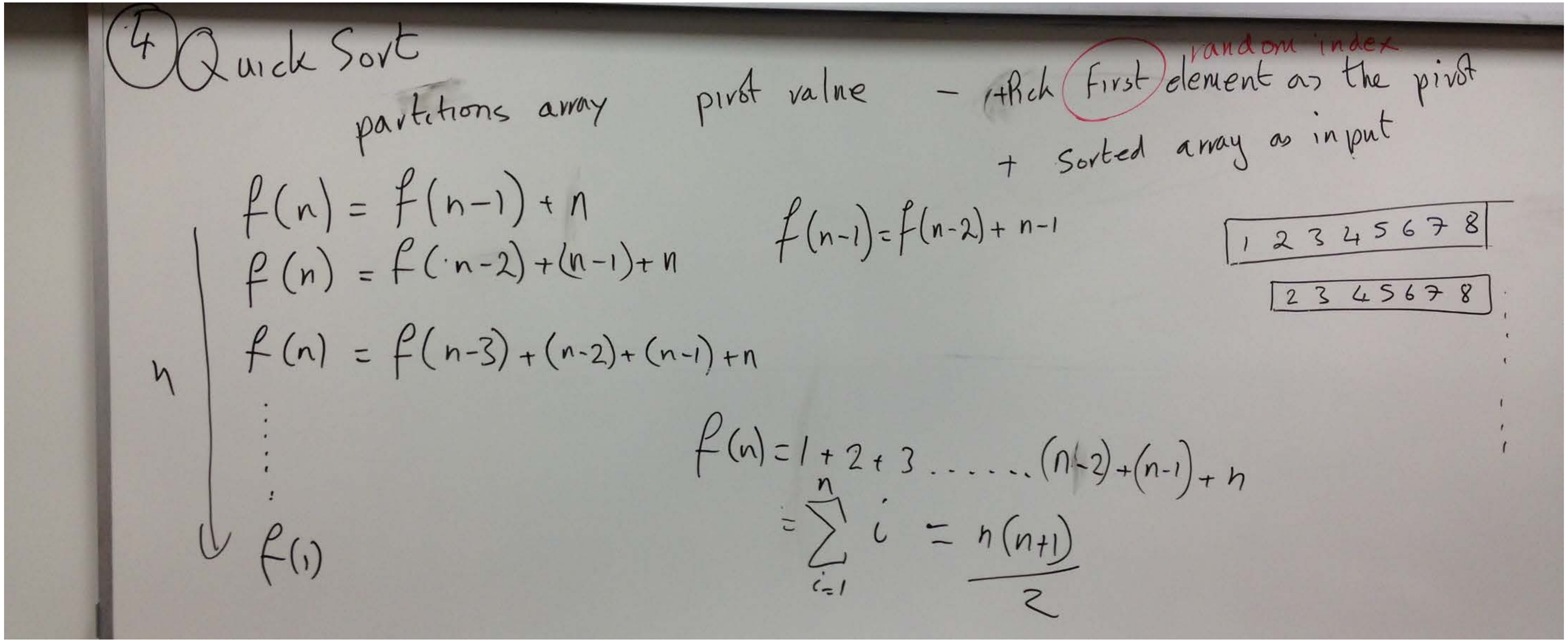
$f(n) = n\,i$

$= n \log_2 n$



total number of steps

area = $n \log_2 n$



④ Quick Sort

partitions array   pivot value   random index

– pick First element as the pivot

+ Sorted array as input

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

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

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

n ⋮

$f(1)$

$$f(n) = 1 + 2 + 3 \ldots\ldots (n-2) + (n-1) + n$$

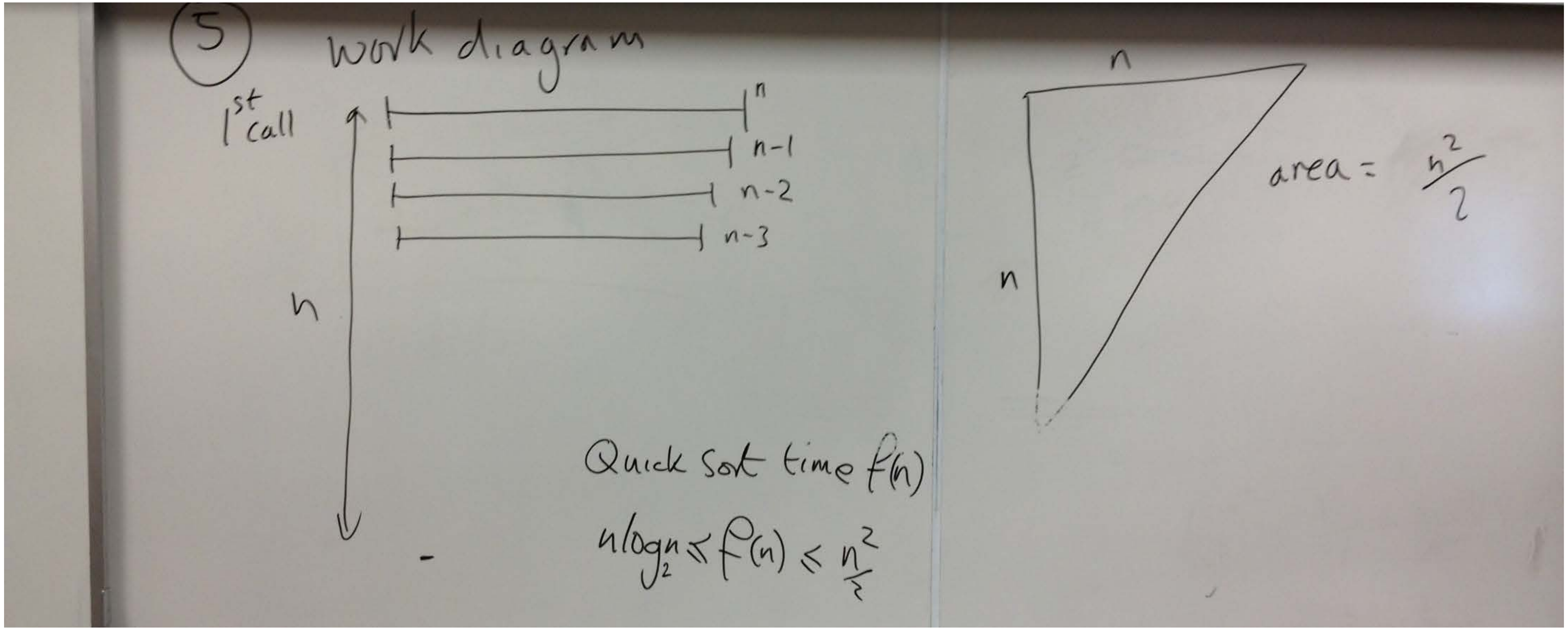
$$= \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

1	2	3	4	5	6	7	8

2	3	4	5	6	7	8

⋮

5 work diagram



area = $\frac{n^2}{2}$

Quick Sort time $f(n)$

$n\log_2 n \leq f(n) \leq \frac{n^2}{2}$