

Algo ① Find max. element in a $< \infty$ seq.

```

procedure max ( $a_1, a_2, \dots, a_n$  : integers)
max :=  $a_1$ 
for  $i := 2$  to  $n$ 
    if  $\text{max} < a_i$  then  $\text{max} := a_i$ 
    {max is the largest element}
  
```

Pseudo
Code

Properties ① Input: (seq. of integers)

② Output: (the largest integer in the seq.)

③ Definiteness: Each step is precisely defined ✓
 as only assignments, a finite loop & conditional statement occur

④ Correctness: In order to show the correctness we must show when algo terminates the value of the variable max equals the maximum of the terms of the seq.
 Note initial value is the first term a_1 as successive terms are examined max is updated to the value of the term if the term exceeds the max. of the terms previously examined. \therefore When all the terms are examined $\text{max} =$ the value of the largest term.

⑤ Finiteness: The algo uses finite no. of steps.

⑥ Effectiveness: The algo can be carried out in $< \infty$ amount of time as each step is either a comparison or an assignment.

⑦ Generality: The algo is general as it can be used to find max. of any $< \infty$ seq. of integers.

Searching Algos

① Linear Search or sequential search algo

procedure linear search (x : integer, a_1, a_2, \dots, a_n : distinct integers)

$$\dot{\lambda}_i = 1$$

while ($i \leq n$ and $x \neq a_i$)

$$i := i + 1$$

if $i \leq n$ then location := i

else location := 0

(location is the subscript of the term that equals to x or is 0 if x is not found)

② **Binary Search** can be used when the list has terms in order of increasing size (or lexicographic in case of words).

Example Search for 19 in the list

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22
 ↑ first split

1 2 3 5 6 7 8 10 12 13 15 16 18 19 20 22

As $10 < 19$ choose other disks & split \uparrow

12 13 15 16

18 19 20 22

As $16 < 19$ Choose other list

18 19

20 22

As 19 is ~~18~~ \leq largest term of first list choose first list

18 19
 ↑

18 19

as $18 < 99$ choose second list

Search is narrowed down to 1 term

comparision is made 1a is located as
14th term

In general,

To search x in a_1, a_2, \dots, a_n

where $a_1 < a_2 < \dots < a_n$

begin by comparing x with middle term of the seq. a_m , $m = \lfloor \frac{n+1}{2} \rfloor$

The Binary Search Algo

```
procedure binarysearch ( $x$ : integer,  $a_1, a_2, \dots, a_n$ :  
                        increasing integers)  
   $i := 1$  {  $i$  is left endpoint of search interval }  
   $j := n$  {  $j$  is right endpoint of " " }  
  while  $i < j$   
    begin  $m := \lfloor \frac{i+j}{2} \rfloor$   
    if  $x > a_m$  then  $i := m+1$   
    else  $j := m$   
  end  
  if  $x = a_i$  then location :=  $i$   
  else location := 0
```

Sorting Algo.

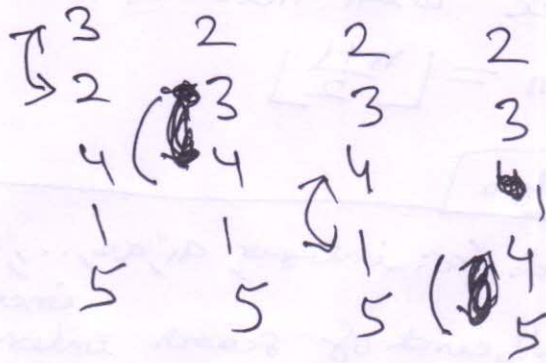
~400 pages 15 different sorting Algo's

Bubble sort

```
procedure bubblesort ( $a_1, a_2, \dots, a_n$ : real nos. with  
                         $n \geq 2$ )  
  for  $i := 1$  to  $n-1$   
    for  $j := 1$  to  $n-i$   
      if  $a_j > a_{j+1}$  then interchange  
         $a_j$  &  $a_{j+1}$   
  {  $a_1, \dots, a_n$  is in increasing order }
```

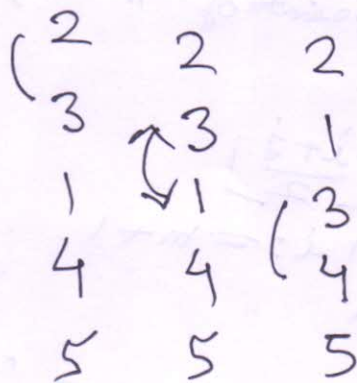

Example 1: Sort 3 2 4 1 5 into increasing order
using bubble sort

First pass

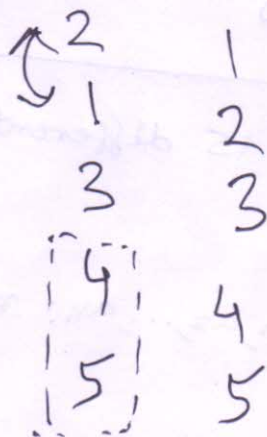


(pair and
correct
order)
↕ interchange

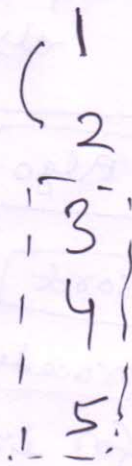
Second pass



3rd pass



nth pass



Insertion sort

procedure insertion sort (a_1, a_2, \dots, a_n : real nos with $n \geq 2$)

for $j := 2$ to n

begin

$i := 1$

while $a_j > a_i$

$i := i + 1$

$m := a_j$

for $k := 0$ to $j - i - 1$

$a_{j-k} := a_{j-k-1}$

$a_i := m$

end { a_1, a_2, \dots, a_n are sorted}

Example Use insertion sort to put elements 3, 2, 4, 1, 5 in increasing order

1 First compare 2 & 3, as $3 > 2$ place 2 at 1st place

2 3 4 1 5
↑

Compare 4 with 2 & with 3

$4 > 2$ & $4 > 3$

4 is placed at 3rd position

list is 2 3 4 1 5

↑
Compare $1 < 2$, $1 < 3$, $1 < 4$ $\therefore 1$ is placed at 1st position.

1 2 3 4 5
↑

as $5 > 4$ it goes to end position

1 2 3 4 5 sorted list

Greedy Algo

That makes best choice at each step

Iteration 1

$$N = 10, S = 1$$

$$i = 1$$

$$S < N$$

$$i = i + 1$$

$$N = N - S$$

$$N = 10 - 1 = 9$$

$$N = 9 - 1 = 8$$

$$N = 8 - 1 = 7$$

End of iteration 1

Example

Use iteration 1 to find minimum value of N in the following array

First compare $2 < 3$, $3 < 5$, $5 < 7$

$2, 3, 5, 7$

↑

Compare N with S and if $N > S$ then $N = N - S$

$$N > S$$

$$N = 10 - 2 = 8$$

↑

Compare $1 < 2$, $2 < 3$, $3 < 5$, $5 < 7$

Iteration 2

$$N = 8 - 2 = 6$$

↑

Compare $1 < 2$, $2 < 3$, $3 < 5$, $5 < 7$

$$N = 6 - 2 = 4$$

Big-O-notation

Order of ∞ - G. H. Hardy— f & g are fns from \mathbb{Z} or \mathbb{R} to \mathbb{R} $f(x) = O(g(x))$ if \exists constants c & k s.t.

$$|f(x)| \leq c |g(x)| \text{ whenever } x > k$$

 \rightarrow Borel const.**Example 1** $f(x) = x^2 + 2x + 1 = O(x^2)$

$$\text{When } x > 1 \Rightarrow x < x^2 \text{ \& } 1 < x^2$$

$$\Rightarrow f(x) = x^2 + 2x + 1 < x^2 + 2x^2 + x^2 = 4x^2$$

$$\therefore c = 4 \text{ \& } k = 1$$

$$\Rightarrow f(x) = O(x^2)$$

Alternat. When $x > 2$ $2x < x^2$ & $1 \leq x^2$

$$\therefore f(x) = x^2 + 2x + 1 \leq x^2 + x^2 + x^2 = 3x^2$$

$$c = 3 \text{ \& } k = 2$$

Note $f(x) = O(x^3)$ or $f(x) = O(x^2 + x + 1)$

$$\text{Also } x^2 = O(x^2 + 2x + 1)$$

$$x^2 < x^2 + 2x + 1 \text{ whenever } x > 1$$

$$c = 1 \text{ \& } k = 1$$

Example 2 $7x^2 = O(x^3)$

$$\text{When } x > 7 \Rightarrow 7x^2 < x^3 \therefore c = 1 \text{ \& } k = 7$$

$$\therefore 7x^2 = O(x^3)$$

Altus: When $x > 1$ $7x^2 < 7x^3 \therefore c = 7 \text{ \& } k = 1$

Example 3 Show that $n^2 \neq O(n)$

We need to show that no pair of const's. c & k exist
s.t. $n^2 \leq cn$ whenever $n > k$

Observe that when $n > 0$, $n^2 \leq cn$ gives $n \leq c$

$n \leq c$ cannot hold for all $n > k$ if we choose
 $k = \max\{k, c\}$ it is not true
 $n \leq c$

Example 4 We know $7x^2 = O(x^3)$ so it is true that
 $x^3 = O(7x^2)$

□ We need to find c & k s.t. $x^3 \leq c \cdot 7x^2$
where $x > k$

$x \leq 7c$ $\forall x > k$ no such c exist.

as x can be made arbitrarily large.

$\therefore x^3 \neq O(7x^2)$

Properties

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

$\forall a_i \in \mathbb{R}$

$$f(x) = O(x^n)$$

□ If $x > 1$ we have

$$\begin{aligned} |f(x)| &= |a_n x^n + \dots + a_1 x + a_0| \\ &\leq |a_n| x^n + |a_{n-1}| x^{n-1} + \dots + |a_1| x + |a_0| \\ &= x^n \left(|a_n| + \frac{|a_{n-1}|}{x} + \dots + \frac{|a_1|}{x^{n-1}} + \frac{|a_0|}{x^n} \right) \\ &\leq x^n (|a_n| + |a_{n-1}| + \dots + |a_0|) \end{aligned}$$

$$\therefore c = \sum_{i=0}^n |a_i| \quad \text{whenever } x > 1$$

$\& k = 1$

$$\therefore f(x) = O(x^n)$$



Example

$$1+2+3+\dots+n = O(n^2)$$

$$\square \because 1+2+\dots+n \leq n+n+\dots+n = n^2$$
$$c=1 \text{ \& } k=1$$

Example

$$n! = O(n^n)$$

$$\square n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n \leq n \cdot n \cdot n \cdot \dots \cdot n = n^n$$

$$c=1 \text{ \& } k=1$$

Take log on both sides

$$\log n! \leq n \log n$$

$$\Rightarrow \log n! = O(n \log n) \text{ taking } c=1 \text{ \& } k=1$$

Example

$$n < 2^n \text{ when } n \text{ is +ve integer}$$

$$n = O(2^n)$$

$$\square \because n < 2^n \therefore \text{ for } c=1 \text{ \& } k=1$$

Take log base 2

$$\log_2 n < n \Rightarrow \log_2 n = O(n)$$
$$c=k=1$$

$$\because \log_b n = \frac{\log n}{\log b} < \frac{n}{\log b} \therefore c = \frac{1}{\log b} \text{ \& } k=1$$

$$\log_b n = O(n)$$

Sum

$$\text{If } f_1(x) = O(g_1(x)) \text{ \& } f_2(x) = O(g_2(x))$$

$$\text{then } (f_1+f_2)(x) = O(\max |g_1(x)|, |g_2(x)|)$$

$$\square |f_1(x)| \leq c_1 |g_1(x)| \text{ \& } |f_2(x)| \leq c_2 |g_2(x)|$$
$$x > k_1 \quad \quad \quad x > k_2$$

$$(f_1+f_2)(x) = |f_1(x) + f_2(x)| \leq |f_1(x)| + |f_2(x)|$$

$$\leq c_1 |g_1(x)| + c_2 |g_2(x)|$$

when $x > \text{both } k_1 \text{ \& } k_2$

$$\leq c_1 |g(x)| + c_2 |g(x)| = (c_1 + c_2) |g(x)|$$

$$c = c_1 + c_2 \text{ \& } g(x) = \max \{ |g_1(x)|, |g_2(x)| \}$$
$$x > \max \{ k_1, k_2 \}$$

Cor. $f_1(x) = O(g(x))$ & $f_2(x) = O(g(x))$

$$\square (f_1 + f_2)(x) = O(g(x))$$

$$\therefore \max\{g(x), g(x)\} = g(x) \quad \square$$

Th. $f_1(x) = O(g_1(x))$ & $f_2(x) = O(g_2(x))$

$$\text{then } (f_1 f_2)(x) = O(g_1(x) g_2(x))$$

$$\square |f_1 f_2(x)| = |f_1(x)| |f_2(x)| \leq c_1 |g_1(x)| \cdot c_2 |g_2(x)|$$

$$\leq c_1 c_2 |(g_1 g_2)(x)|$$

$$|f_1 f_2(x)| \leq c |g_1(x) g_2(x)| \quad \text{"} c \text{" } \leftarrow k = \max\{k_1, k_2\}$$

Example 1

$$f(n) = 3n \log n! + (n^2 + 3) \log n = O(n^2 \log n)$$

$$\square \log n! = O(n \log n) \quad \& \quad 3n = O(n)$$

$$\therefore 3n \log n! = O(n^2 \log n)$$

now

$$\underline{n^2 + 3 < 2n^2} \text{ when } n > 2 \Rightarrow (n^2 + 3) \log n \leq 2n^2 \log n$$

$$\therefore n^2 + 3 = O(n^2) \quad \checkmark$$

$$c=2 \quad k=2$$

$$\therefore \underline{n^2 + 3 \log n = O(n^2 \log n)} \quad \leftarrow$$

$$\therefore f(n) = O(n^2 \log n) \quad \checkmark \quad \square$$

Example

$$f(x) = (x+1) \log(x^2+1) + 3x^2 = O(x^2)$$

$$x+1 = O(x) \quad x^2+1 \leq 2x^2 \text{ when } x > 1$$

$$\therefore \log(x^2+1) \leq \log(2x^2) = \log 2 + 2 \log x$$

$$\therefore \text{for } x > 2 \quad \log(x^2+1) = O(\log x) \leq 3 \log x$$

$$\therefore (x+1) \log(x^2+1) = O(x \log x)$$

$$3x^2 = O(x^2)$$

$$\therefore f(x) = O(\max(x \log x, x^2))$$

$$\therefore x \log x \leq x^2 \text{ for } x > 1$$

$$f(x) = O(x^2) \quad \square$$

Big Ω -notation

$$f \text{ and } g: \mathbb{Z} \text{ or } \mathbb{R} \rightarrow \mathbb{R}$$

$$f(x) = \Omega(g(x)) \text{ if } \exists \text{ +ve constants } c \text{ \& } k \text{ s.t.}$$

$$|f(x)| \geq c|g(x)| \text{ whenever } x > k$$

Example

$$f(x) = 8x^3 + 5x^2 + 7 = \Omega(x^3) \text{ " } g(x)$$

$$\because 8x^3 + 5x^2 + 7 \geq 8x^3 \text{ } \forall \text{ +ve reals}$$

$$\text{i.e., } g(x) = O(8x^3 + 5x^2 + 7)$$

Standard Ref. fns.

$$x^n, n > 0 \text{ or } e^x, c > 0$$

Big Θ -fn.

$$f \text{ and } g: \mathbb{Z} \text{ or } \mathbb{R} \rightarrow \mathbb{R}$$

$$f(x) = \Theta(g(x)) \text{ if } f(x) = O(g(x))$$

$$\text{ \& } f(x) = \Omega(g(x))$$

Ref. fns

$$x^n, e^x, \log x, \dots$$

Example

$$1 + 2 + 3 + \dots + n = \Omega(n^2)$$

$$\therefore 1 + 2 + 3 + \dots + n = \Theta(n^2)$$

$$\square \text{ We know that } 1 + 2 + \dots + n = O(n^2)$$

$$1 + 2 + 3 + \dots + n \geq \left\lfloor \frac{n}{2} \right\rfloor + (\left\lfloor \frac{n}{2} \right\rfloor + 1) + \dots + n$$

$$\geq \left\lfloor \frac{n}{2} \right\rfloor + \dots + \left\lfloor \frac{n}{2} \right\rfloor$$

$$= (n - \left\lfloor \frac{n}{2} \right\rfloor + 1) \left\lfloor \frac{n}{2} \right\rfloor$$

$$\geq \left(\frac{n}{2}\right) \left(\frac{n}{2}\right) = \frac{n^2}{4}$$

(ignore first half
& sum only the terms
 $> \left\lfloor \frac{n}{2} \right\rfloor$)

$$\therefore 1 + 2 + \dots + n = \Omega(n^2) \therefore f(n) = \Theta(n^2)$$

$$f(x) = \Theta(g(x))$$

$$c_1 |g(x)| \leq |f(x)| \leq c_2 |g(x)| \quad x > k$$

$$f(x) = O(g(x)) \text{ \& } f(x) = \Omega(g(x))$$