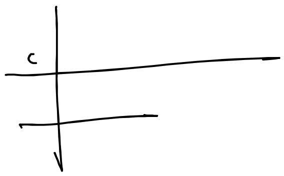
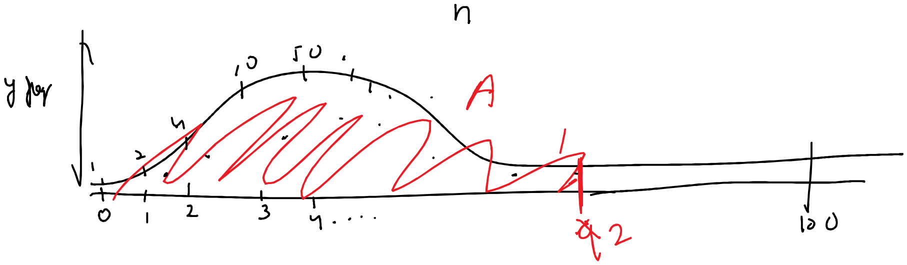


O

Ω

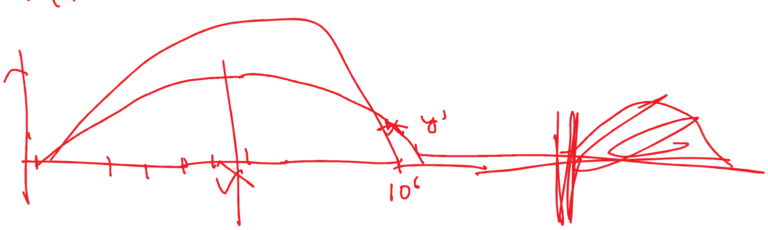
$f(n) = 2 + 3\underset{\hookrightarrow n}{l} = 2+3n \rightarrow \underline{O(n)}$

n



$\underline{A}$

Total area



{ ______________ }

$K + C \cdot \underline{\log n} = f(n)$

$\hookrightarrow O(\quad n$

$f(n) = 1 + 2(1 + \log \frac{n}{2})$

n

$f(n) = 1 + 2n \rightarrow O(n)$

```
while (i<=n){
   i++;
}
```

i<=n

```
w( ){
  i = i x 2;
```

```
while (i<=n){
   i= i+2
}
```

$f(n) = 1 + 2 \cdot l$

```
w( ){
  i = i x 3 }
```

```
while (   ){
   i = i+2
   i = i+3
}
```

$1 + 2l = f(n)$

```
w( ){
   i = i x 2
   i = i x 3
}
```

```
while ( ) {
```

```
w( ) {
```

l	i
1	1
2	K
3	K^2
l	K^{l-1}

$K^{l-1} <= n$

i = i + 3   f(n)
}
while ( ) {
i = i + k }

i × 3
}
w( ) {
i = i × K
}

$k^{l-1} <= n$

$l - 1 = \log_k n + 1$

$f(n) = 1 + 2(1 + \log_k n)$

$O(\log_k n)$

$\log_a c$

$\frac{a}{b} = 0$

```
while (n > 0) {
   n = n / 2;
}
```

$n = 2^{l-1}$

$\log_2 n = l$

l	n
1	n
2	$n/2$
3	$n/2^2$
4	$n/2^3$
l	$n/2^{l-1}$

$n/2^{l-1} \Rightarrow 0 = 1$

```
for( i=0; i^2 <= n; i++) {
   Syso(i);
}
```

$\sqrt{n} = n^{1/2}$

$n^{-1} = \frac{1}{n}$

$(\sqrt{n} + 1) \cdot c_1 + c_2 = f(n)$

$i=1$ | 2 | 3 | 4 | ... | n

$j = 1$ to 1 | $j=1$ | 1 | 2 | 3

$k \to 1000$ | 1000 | 1000 | 1000 | 1000 | 1000

1K | 2K | 3K | $n \cdot 1K$

$$1K(1 + 2 + 3 + \dots n) \Rightarrow 1000 \cdot \frac{n \cdot (n+1)}{2}$$

$$O(n^2)$$

$i=1$ | $i=2$ | $i=3$ | $i=4$ | $i=n$

$j=1$ | $j = 1$ to 2^2 | $j = 1$ to 3^2

$n/2$ | 1: $n/2$ | 2: $n/2$ | 3: $n/2$ | 4: $n/2$ | $3^2 \cdot n/2$ | $4^2 \cdot n/2$ | $n^2 \cdot n/2$

$$n/2 + 4 \cdot n/2 + 3^2 \cdot n/2 \dots + n^2 \cdot n/2$$

$$n/2 (1 + 2^2 + 3^2 + \dots n^2)$$

$$O(n^4) \qquad n/2 \cdot \frac{n \cdot (n+1)(2n+1)}{6}$$

$$\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots \frac{1}{n}\right)$$

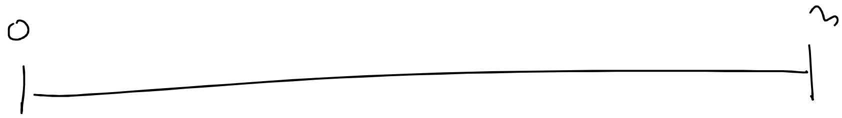
$\log n$

$i = 1, 2, 3 \dots n$

$$n + n/2 + n/3 + \dots n/n$$

$$n\left(1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\cdots\frac{1}{n}\right) = n\log n$$

$$\int \frac{1}{x}$$



n/K

$O(1)$ $O(\log n)$ $O(\sqrt{n})$, $O(n)$, $O(n\log n)$

$O(n^2)$, $O(n^2\log n)$, $O(n^3)$, $O(c^n)$, $O(n!)$, $O(n^n)$

$\cancel{O(n^c)}$

$c^n \le n!$, n^n

$(c\cdot c\cdot c\cdot \quad c)$ 5?

$(1\cdot 2\cdot 3\cdot\, c\cdot (c+1)\,(c+2)\cdots n)$

1) 4.4 H2

① Max Subarray Sum

Q Max Sub array sum

$n \le 10^5$

$O(n^3)$

$10^{15} / 10^9 = 10^6$

$O(n^2)$

$O(10^{10})$

$10^{10}/10^9 \approx 10$

$O(n)$

10^5

	W	Best
Bubble	$O(n^2)$	$\Omega(n^2) \to \Omega(n)$
Selec	$O(n^2)$	$\Omega(n^2)$
Ins	$O(n^2)$	$\Omega(n)$

Given : 1 * x^n+2*x(n-1)+3*x(n-2)... + nx;
solve, return number
n=3;
x=7;

$1\cdot(7)^3 + 2\cdot(7)^2 + 3\cdot 7^1$

Q $\to$ $1\cdot x^n + 2\cdot x^{n-1} + 3\cdot x^{n-2} + 4\cdot x^{n-3} + \dots + nx^1$

$n \to 3$, $x = 7$

$O(n)$

$7^3 + 2\cdot 7^2 + 3\cdot 7^1$

$343 + 2\times 49 + 21$

98

462

$n \to 2$

$x \to 1$

$1\cdot(1)^2 + 2(1)^1$

$\Rightarrow$

x^n

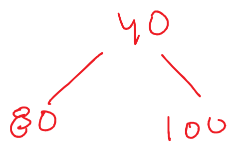
x^5

x^4 $\quad x^2 \quad$ x^5

$x \times x$

$x^2 \times \quad x^2 \times x$

$x^5 \times x^5 \quad x^5$



(Key, Value) →

unique

array → 0 . n-1

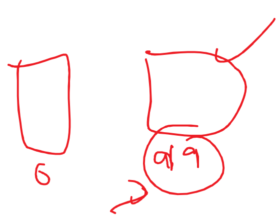
A → 10
B → 20
C → 15
D → 10,
E → 20

Key →	A	B	C	D	E
Value	10	20	15	10	20

Search → O(n), Key & Value
Add → O(1)
Remove → O(n)
Update → O(n)

Has

Search → Key O(1), Update O(1)
Add → O(1)
Remove → O(1)



n×n

int[] arr1 = { 30, 20, 40, 50, 70, 30, 20, 20, 50, 50 };

+∞

int[] arr2 = { ~~50~~, 80, 30, 20, 20, 20, 90, 50, 20 };

✓ 30, 20, 50, 20, 20, 50

Key val

int[] arr1 = { 30, 20, 40, 50, 70, 30, 20, 20, 50, 50 };

int[] arr2 = { ~~50~~, 80, ~~30~~, ~~20~~, ~~20~~, ~~20~~, 90, ~~50~~, 20 };

30, 20, 50, 20, 20, 50

H.M <key, occ>

Q number of string concatination for string s to be a subsequence of string t

From
<https://docs.google.com/spreadsheets/d/14W2tmEZ1X2pr1e9kVV1cu74wBapshhURPyUBdKx0OQM/edit#gid=1635212751>

2.
t → cbcd → n   n log n
s → catsbanddogs → m   n log m, m log n

catsband dogs cats band dot

You are given an array of Integers in no particular order. Write a Program to find the longest possible sequence of consecutive numbers using the numbers from the array.

Target O(n)

Input : [2,12,9,16,10,5,3,20,25,11,1,8,6]

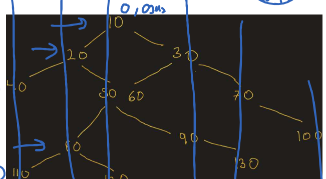
1, 2, 3, 5, 6, 8, 9, 10, 11, 12
16, 20, 25

1, 2, 3, → 3

Output : [8,9,10,11,12]
Input : [15,13,23,21,19,11,16]
Output : [15,16]

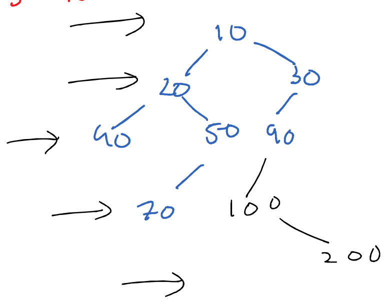
O(n)

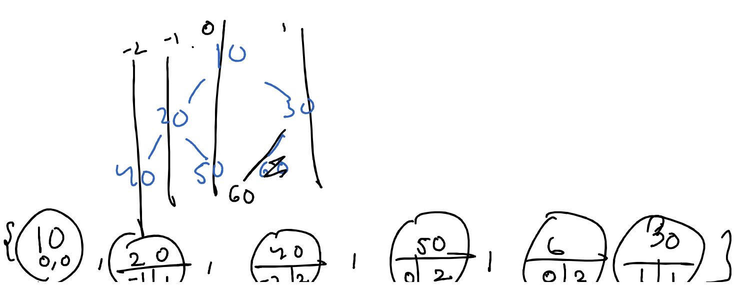
Left
Right
top
Bottom
Vertical
Boundary

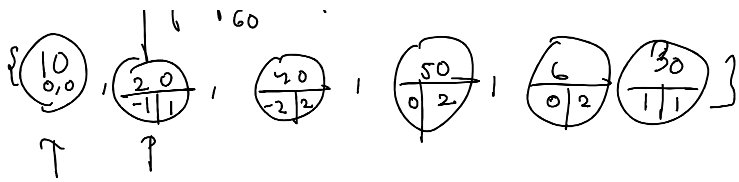


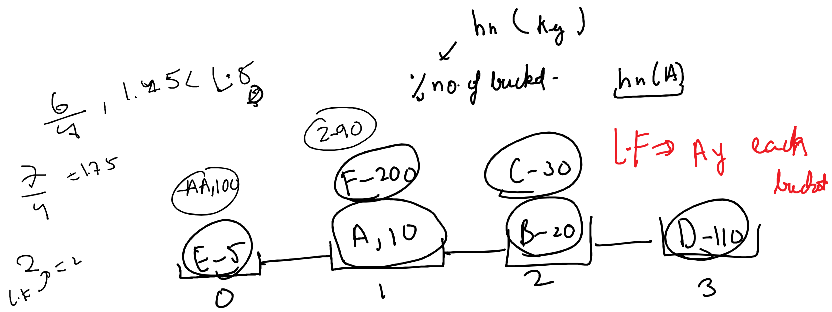
H.M <integer, A.L>

0 → 10, 50, 60, 120
1 → 20, 80
1 → 30, 90
-2 → 40, 110
2 → 70, 130
3 → 100





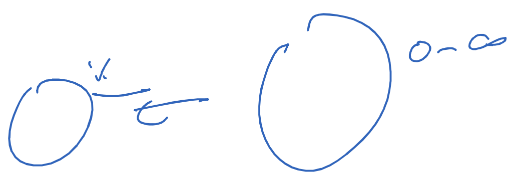




avg size of each bucket ⇐ L.F ?? , collision resolve ?? → chaining.

th ?? , default bucket ↳ 16

Array ?!



1	2	3	~~4~~	5
~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~
~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~

Num = 2 Div = 1,2	Num = 3 Div = 1,2,3	Num = 4 Div = 1,2,3,4			Num = n Div = 1 to n

N => n/2
N=> N/3

n/2+n/3+n/5+n/7+n/11...... +n/n
N log(log(n))

F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F

1, 2, 3, 4, 5, n

200 x 500 x 4 x 2 byte

8,00,0~~00~~ byte

0.8 MB

var1 = → the
v2 →
v3 →
v9999 →

10K

String	n	Operation
""	0	1
"0"	1	1+1
"01"	2	2+1
"012"	3	3+1

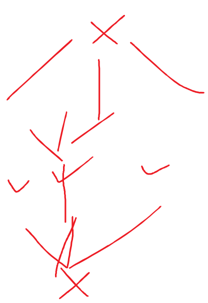
```
String str1 = "Hello"; // Literal
String str2 = "Hello";
String str3 = "Hello";
```

H
IP

10K
Hello

	String	StringBuilder
Create	String str = “hello”; String str = new String(“hello”)	String sb = new StringBuilder(“hello”);
Print	Syso(str)	Syso(sb)
Char at	str.CharAt(index)	str.CharAt(index)
SubString	str.substring(index);	str.substring(index); // return String

Add
Interset
Update
Remove
	X	sb.append(“”);
		sb.insert(index,””); char or string
		sb.setCharAt(index,’’);
		sb.deleteCharAt(idx);
		sb.delete(including,excluding);
		sb.toString(); \\ return to string



1) → B.P.

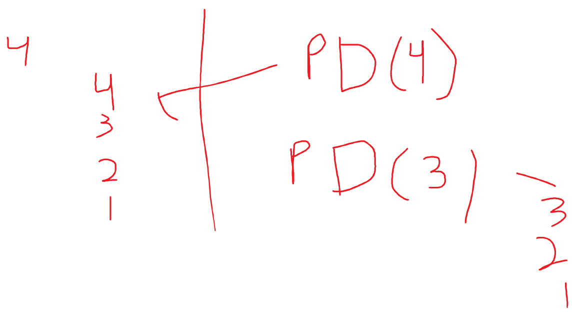
4) → Self.W
Bi soln

1) → B. P. Bi soln

2) → S. P.

3) → Assume

4

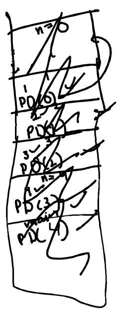


BP : PD(4)
SP : PD(3)

PD(4){
Syso(4)
PD(3)

}

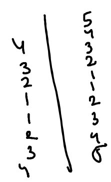
```
public static void PD(int n) {
        System.out.println(n);
        PD(n-1);
}
```

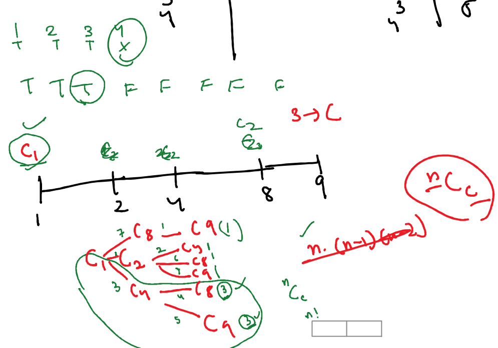


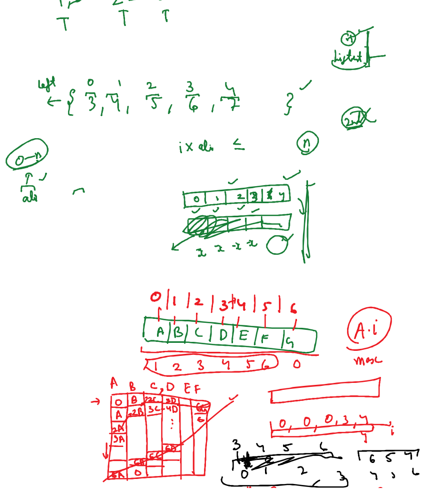
4
3
2
1

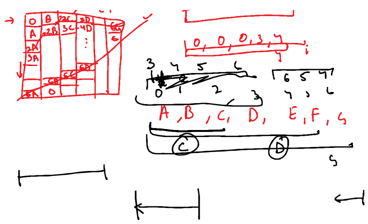
PI (4) → 1 2 3 4

PDI → 4





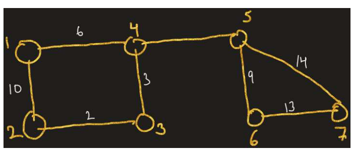




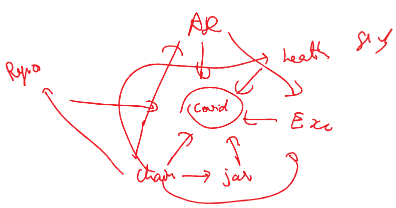
0 1 2 3 4↔5 6
A, B, C, D, E, F, G.

{3, 6, 4, 2, 5, 2} -O O O
0 1 2 3 4 5
0 1 2 3 4
-O

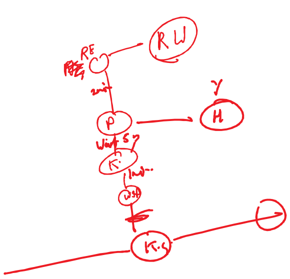
→ {A, B, C, D}

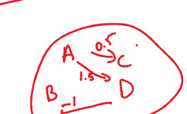


8 edges
7 → vertex / node



2
↳ Weighted ↳ Directed
↳ Non Weighted ↳ Undirected.





N N^2

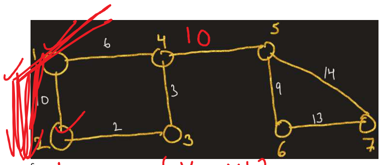
	A	B	C	D
A	0	0	0.5	1.5
B	0	0	0	0
C	0	0	0	0
D	0	-1	0	0

Adjacency matrix

$V \to {}^{V}C_2$
$\to {}^{V}P_2$
$V \cdot (V-1)$

A → B ? : A → {C, D}
B → ~~X~~
C → ~~X~~
D → {B}

→ Adjacency matrix
→ O(N)
A → B



$V_1 \to (V_2, wt)$ 2

$1 \to \{(2,10), (4,6)\}$

$2 \to \{(1,10), (3,2)\}$

$3 \to \{(4,3), (2,2)\}$

$4 \to \{(1,6), (3,3), (5,10)\}$

$5 \to \{(4,10), (6,9), (7,14)\}$

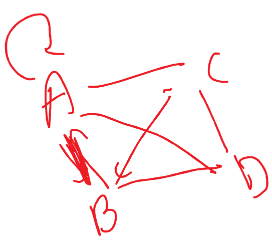
$6 \to \{(5,9), (7,13)\}$

$7 \to \{(6,13), (5,14)\}$

$\langle V_1, N(V) \rangle$

$\langle V_2, wt \rangle$

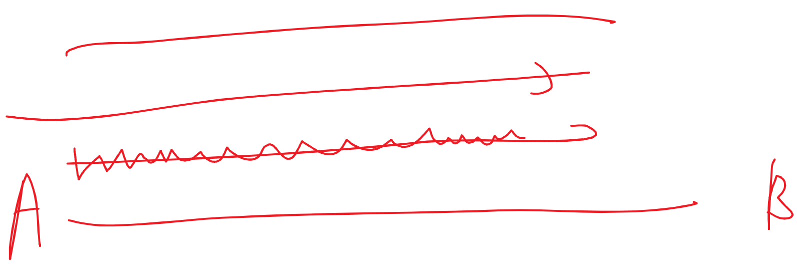
$^4C_2$



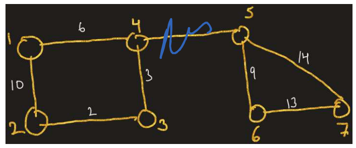
6

$^4P_2$

$\frac{4 \cdot 3}{12}$



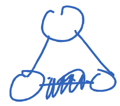
C·B

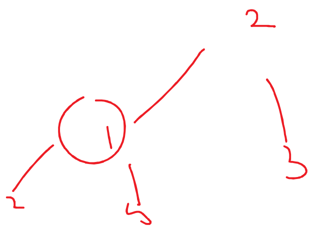


Connected → 1 component

1	2	4	3	5	3	6	7	7

1 2, 4, 3, 5, 6, 7



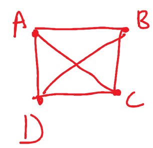


BF Traversal

V_1 — V_2 · · · — V_n

$V+V$

$\simeq O(V)$

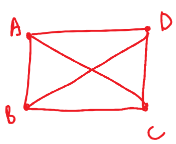


A	B	C	D	C	D	D

$\frac{V(V+1)}{2} + E$

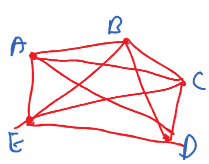
$O(V^2)$

```java
public boolean hasPath2(int src, int dest) {
		boolean ans = false;
		HashSet<Integer> Visited = new HashSet<>();
		Queue<Integer> Q = new LinkedList<>();
		Q.add(src);
		while (!Q.isEmpty()) {
			int curr = Q.poll();
			if (curr == dest) {
				ans = true;
			}
			if (Visited.contains(curr)) {
				System.out.println("Cycle exist");
				continue;
			}
			Visited.add(curr);
			for (int nbr : map.get(curr).keySet()) {
				if (!Visited.contains(nbr)) {
					Q.add(nbr);
				}
			}
		}
		return ans;
	}
```



$O(E+V) \rightarrow$ BFT

$O(E+V)$



A	B	C	D	E	C	D	E	D	E	E

$n \cdot \frac{(n+1)}{2}$

Min spanning tree ← Single src shortest Path

Tree

Kruskal

Prims

Floyd warshall

Union Find

1

PQ min Heap

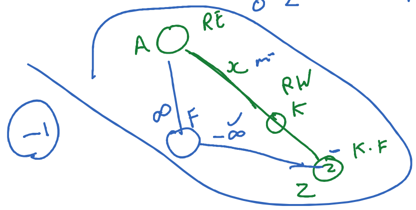
1/0	4/10	2/2	3/5	4/6	5/14	6/19	7/20	7/23

1	2	3	4	6	7
0	2	5	6	19	20

RE

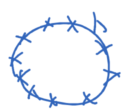
Find 1

1 2 3 4 6 7
0 2 5 6 19 1 20



A. compare B

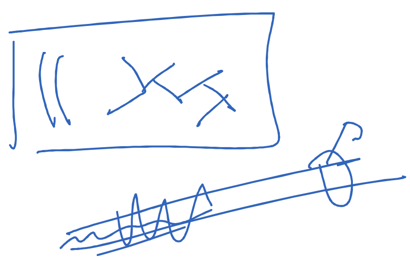
A. wt — B wt



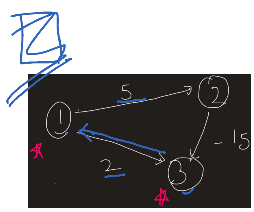
$E \log E \to V^2$

$E \log V^2$

$E \log V$



Bellman Ford



−10

−18

−26

−32