arr → T*
size → size_t
capacity → size_t

int Pay-4 Vector Implementation

~~cap~~ ~~new~~ Insert:(T & val)

capacity:1
size:0

{ size == capacity
capacity *= 2;
auto arr = new T[capacity];
for (int i=0; i<size; ++i)
arr[i] = this->arr[i];
size ++, arr[size-1] = val;
delete[] this->arr;
this->arr = arr;

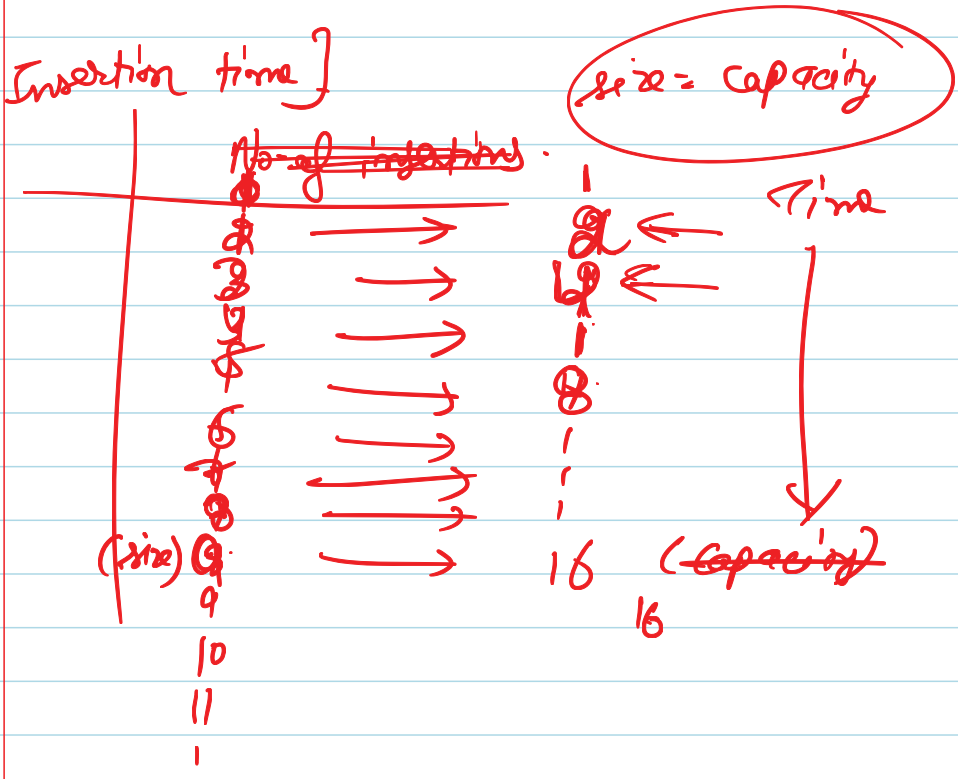
How to access arr of class instead of local variable.

}

```
void pop_back()
{
  assert(size>0);
  size--;
}
```

Insertion time

size = capacity



Insertion time

1 + 2 + 4 + 1 + ... + 8 + 1 + 1 + 1 + 1 + 1 + 16 + 1 + 1 + 1 + 1 + 1 + 1 + 32

8 times (1 1 1 1 1 1 1 1)

16 times (1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)

n

$n \geq 100$
1000

$\leq 2n$ operations $\Rightarrow$ $O(1)$ per element

size--;

```
dstor
{
    delete[] arr;
}
```

Union-Find DSU

n skills

skill $i \rightarrow T_i$

$i \rightarrow +P_i$

Some skills are in a group (x, y)

$G_1 \rightarrow$ i, j, k $\rightarrow$ Time $= \sum t_i$

Power gain $= \sum P_i$

in B time, find the maximum power gain you can achieve.

1 → G1

$n \leq 1000$

$B \leq 10^3$

$T_i \leq 10^3$

$P_i \leq 10^6$

1 → N

→ good form = remove duplicate prime factors of N.

$N = 12, \quad 12 = 2^2 \times 3$

$\rightarrow 2 \times 3$

(M)

better form = multiplication of all divisors of M

→ P.

best form = No. of divisors of P. (⟶)

~~$t \leq 10^2$~~

$t = 1$

$N \leq 10^{15}$

modulo 10^9+7

$\sqrt{n} \rightarrow (p_i, \alpha_i)$

$N = \prod_{i=1}^{k} p_i^{\alpha_i}$

$\sqrt{10^{15}} = 10^8$

ans: 1

$(\alpha_i + 1)$

$12 = 2^2 \times 3 \xrightarrow{M} 2 \times 3 = 6$

P → 6 → {1, 2, 3, 6} → 36

{1, 2, 3, 4, 6, 9, 12, 18, 36} → 9

let good form = $M = \prod_{i=1}^{K} p_i$

Better form = $\prod_{i=1}^{K} p_i^{K-1}$

best form = $(2^{K-1} + 1)^K$ ✓

3

A.W

array, size n

segment size = x

$n > x$

for each segment, find the minimum

Priority queue

(a) Count of minimums = $n - x + 1$

{ ——— }

(b) → max

t = 1

$1 \le n \le 10^6$

$1 \le x \le n$

$1 \le a_i \le 10^9$

Without using segment tree

$T = ?$

(4) vector<string> $N \le 10^5$

$len(a_i) \le 50$

$a_{ij} \in \{'a', 'b', ——— 'z'\};$

$Q \le 10^5$

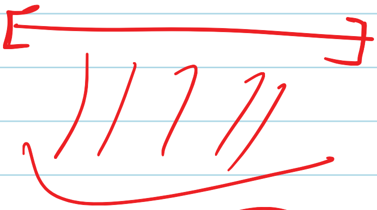
$1 \le L \le R \le N$

L R K

$a_L, a_{L+1}, —— a_R$

$1 \le K \le \sum_{i=L}^{R} size(a_i)$

Copy



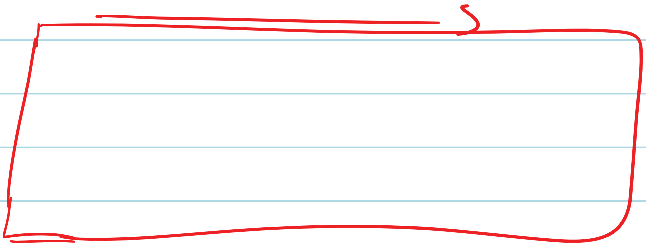
Sort

K^{th}

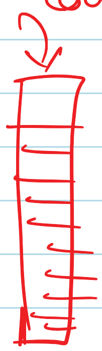
n

$26 \times n$

26



Count.



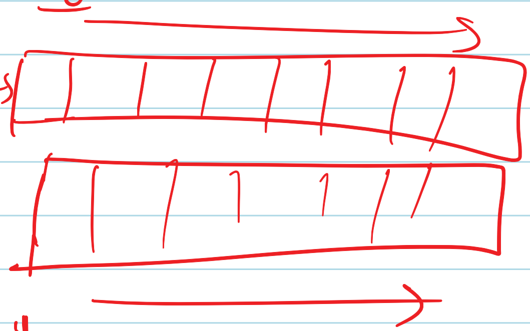
Investor

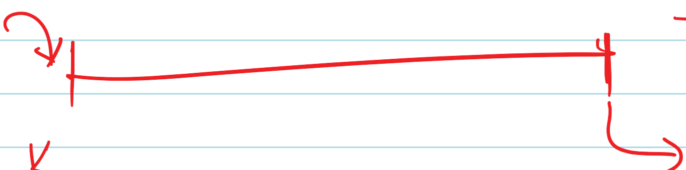
S → Savings

n.

current Value of Stocks

Future Value of Stocks (1 साल)





K

Profit

− Kisa = Profit

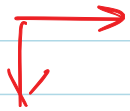
$0 \le n \le 100$

$0 \le$ serving $\le 30,000$

$0 \le$ cur[i][j] ≤ 300

---

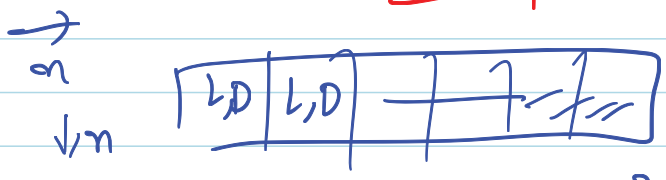
(5) matrix



① (1,1) → (N,N) → count

② (1,1) → (N,N) → print all paths

N: $\sum$ path length of each path $\le 10^9$



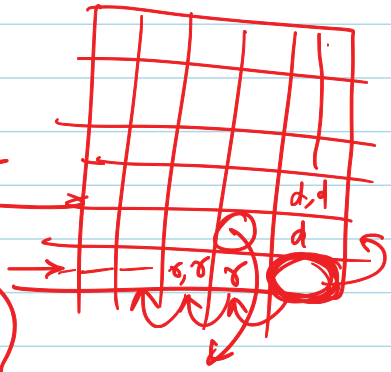
$2n C_n$

$2n$ ($n \times {}^{2n}C_n \le 10^7$)

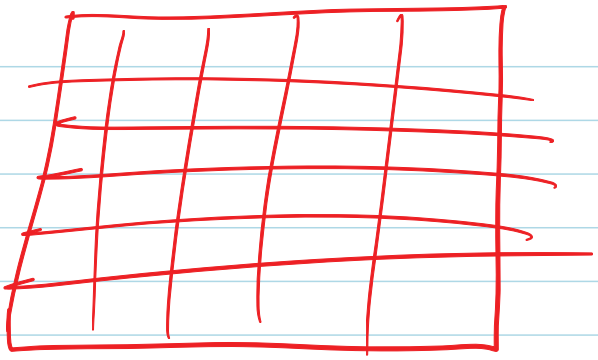
Recursion:

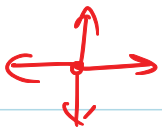
(1,1) → (1,2) → (1,3) → (2,3) - - -

```
int N;
foo (int i, int j)
{
  vector<vector<string>>
  for (i = N-1 to 1)
    for (j to N-1 to 1)
    {
      dp[i][j] = (dp[i+1][j][k] ∪
      dp[i][j]   dp[i][j+1][k] + 'd';
```



(6) H.W.





$m[i][j] = a_{ij}$

$1 \le a_{ij} \le 10^9$

$\forall\ i \in \{1, 2, \dots N\}$

$j \in \{1, 2, \dots M\}$

$N \times M$.

Find the ~~length~~ of longest increasing path.

P: a_{ij}, ~~$a_{i+1,j}$~~ $a_{x_1}\ a_{x_2}\ a_{x_3}$ — — —

$a_{x_1} < a_{x_2} < a_{x_3}$ — —

→ $dp[i][j]$ = len ——— starting from $[i][j]$

⇒

$dp[i][j]$ = pseudo code

$1 \le T \le 1$

$1 \le N, M \le 5 \times 10^3$

dp + recursion

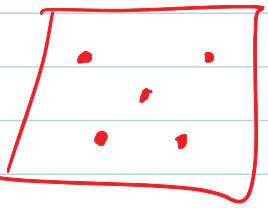
{ value

i

j

}

α starting



for each starting point

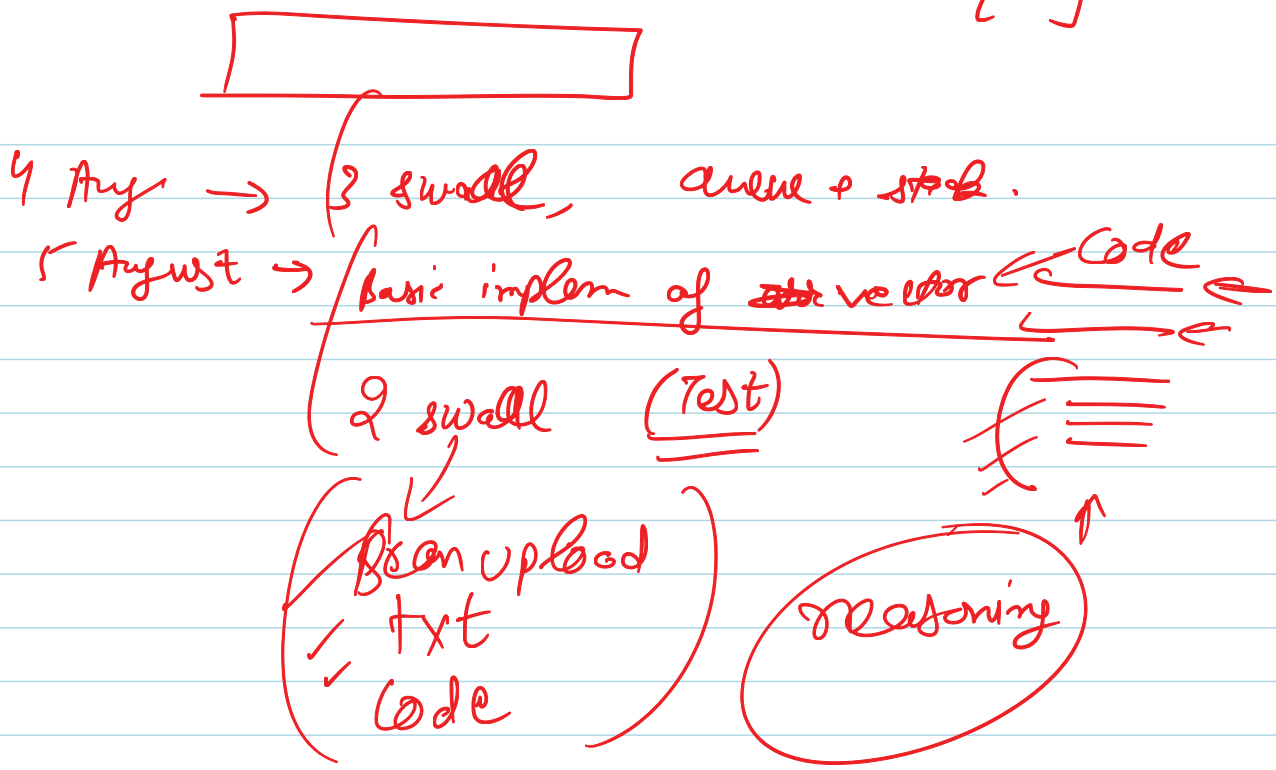
{

——

——

}

dp + recursion (memoization)


[ ]
4 Aug →
Augst →
queue + stack.
Basic implem of vector
Code
2 small
(test)
Bcan upload
txt
code
reasoning