

Tug of war : Divide elements of array in two parts such that diff. of sum of array should be min.

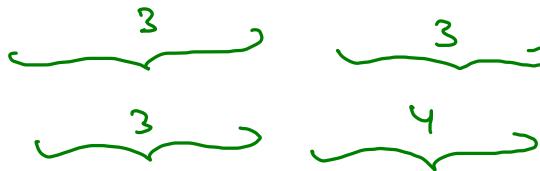
i) Diff. should be min

ii) No. of Element should be equal  
size diff = 0

or may have size diff = 1

Total Elements → 6 size diff = 0

→ ↗



④ = 0  
②

size diff = 0

size diff = 1

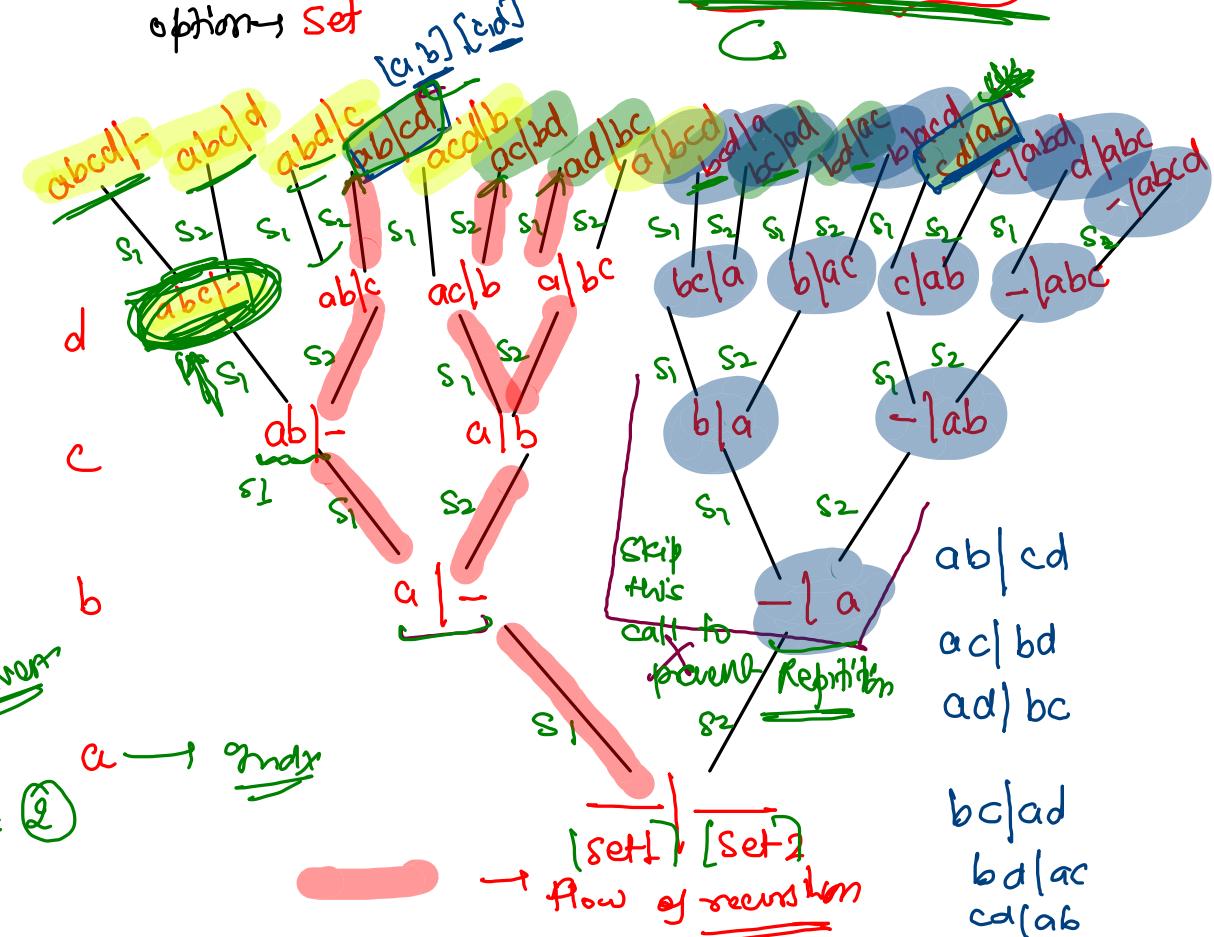
Min diff → ③ Total Element = h  
 $\frac{h+1}{2}$  = ③ Elements in a single set  
 Elements in a single set →  $\frac{n+1}{2}$   
 $\frac{4+1}{2} = 2$

a → qndx

Element → a, b, c, d → split in two parts.

Level → Elements  
options → Set

Valid → size diff = -1, +1, 0



## Pattern Matching

String → graph trees graph

Pattern → pep

o/p →

p → graph  
e → trees

graph trees graph  
p e p

String → ~~mzaddytzaddy~~

Pattern → ~~a b cb~~

m z a d d y t z a d d y

String → m z a d d y t z a d d y ① a → m, b → zaddy c → t  
a b c b

String → m z a d d y t z a d d y ② a → mz, b → addy c → tz  
a b c b

String → m z a d d y t z a d d y ③ a → mza, b → ddy, c → tza  
a b c b

String → m z a d d y t z a d d y ④ a → mzd, b → dy, c → tzd  
a b c b

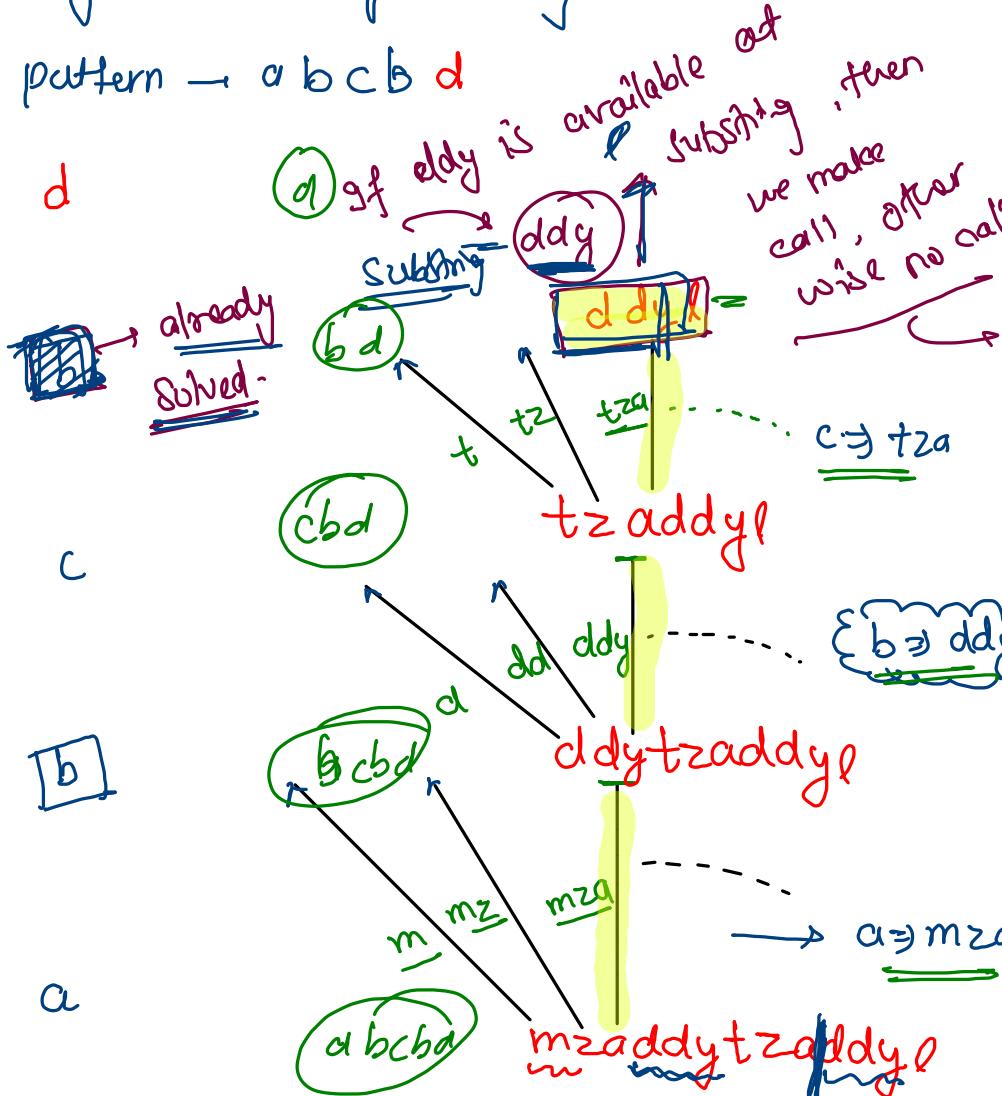
String → m z a d d y t z a d d y ⑤ a → mzadd, b → y, c → tzadd  
a b c b

pattern,  
String

string → mzaddytzaddyo

level → character  
of pattern

option → substring  
of  
String



a → mza  
b → ddy  
c → tza

By default  
pattern's character  
have mapping  
with Empty String.

a → "  
b → "  
c → "

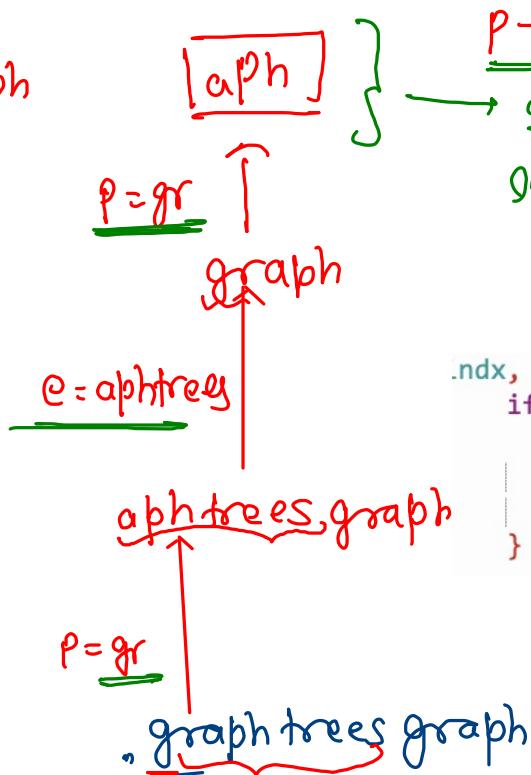
pattern → pep

string → graphtrees graph

P

e

P



$P \rightarrow g$  e → raphtrees

$P \rightarrow gr$ , e → alphatreeg

gf we are at final

level then question string

have no character.

```
int idx, HashMap<Character, String> map, String str;
if(idx == pattern.length()) {
    if(str.length() == 0)
        System.out.println(asf + ".");
}
return;
```

## Word Break - I

dictionary → {i, like, pep, coding, pepcoding, pepper,  
HashSet<String> eating, man, go, mango, in}

String → ilikepeppereatingmangoinpepcoding

i like pepper eating man go in pep coding

i like pepper eating man go in pepcoding

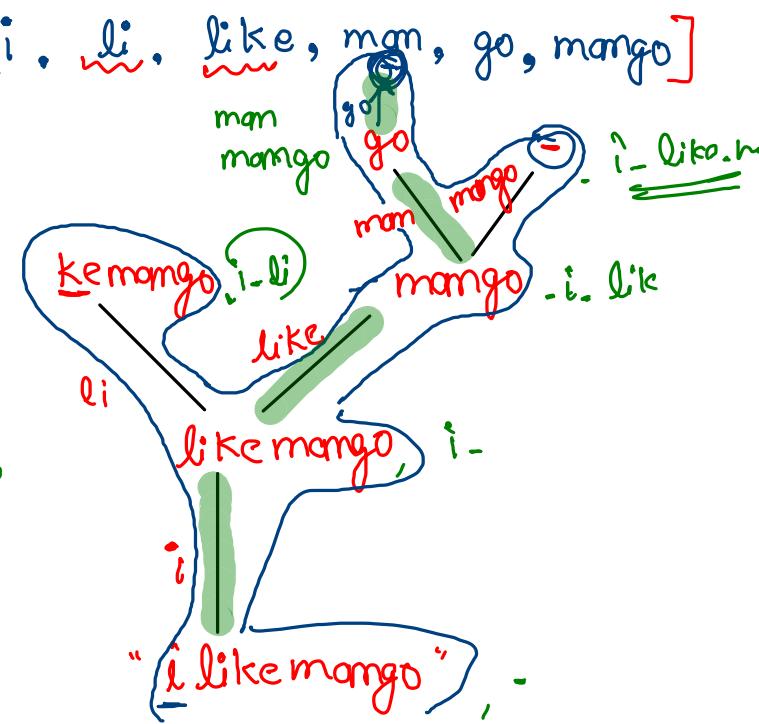
i like pepper eating mango in pep coding

i like pepper eating mango in pepcoding

## Requirement to Solve →

- ① Hashset<String> for Dictionary
  - ② String str
  - ③ String answer so far

l  
~~li~~  
lik  
~~like~~  
likem  
likema  
likeman  
likemong  
likemango  
~~i~~  
il  
ili  
ilik  
i like  
i likem  
i likema  
i likeman  
i likemong



i like mom go  
i like mango

Remove Invalid Parenthesis: String  $\rightarrow$  ~~( ) ( )~~ Invalid  $\rightarrow$  ~~) ) ( (~~  $\rightarrow$  ④

$( ) ( ) \rightarrow$  ~~( ( ) )~~ valid

$( ) ( ) \rightarrow$  ~~( ) ( )~~ empty one for any<sup>a</sup>

$( ) ( ) \rightarrow$  ~~( ) ( )~~

String  $\rightarrow$   $( ) ( ) ( ) ( )$

$( ) ( ) ( ) \rightarrow$  ~~( ( ) ) ( )~~

$( ) ( ) ( ) \rightarrow$  ~~( ) ( ) ( )~~

$( ) ( ) ( ) \rightarrow$  ~~( ) ( ) ( )~~

$(_1)_2)_3(_7)_8)_9)_0$

<del>(8)</del> <sub>9</sub>
<del>(7)</del> <sub>10</sub>
<del>)</del> <sub>11</sub>
<del>)</del> <sub>12</sub>
<del>(</del> <sub>13</sub>

String  $\rightarrow$   $(_1)_2)_3)_4)_5)_6)_7)_8)_9)_0$

Find no. of invalid parenthesis ??

no. of remaining  
bracket = 2

there are 2  
invalid bracket

$\rightarrow$  to make it valid  
we have to remove  
Exactly 2 bracket

## How to find min Number of Removal? →

String →

(<sub>1</sub>)<sub>2</sub> (<sub>3</sub>)<sub>4</sub> )<sub>5</sub> )<sub>6</sub> ( <sub>7</sub> ( <sub>8</sub> )<sub>9</sub> )<sub>10</sub>

Steps:

① stack < character>

② opening bracket

↳ push in stack

closing bracket

→ top. opening

↳ pop.

→ otherwise

push

(<sub>8</sub>)<sub>9</sub>  
(<sub>7</sub>)<sub>10</sub>  
)<sub>6</sub>  
)<sub>5</sub>  
(<sub>3</sub>)<sub>4</sub>  
(<sub>1</sub>)<sub>2</sub>

remaining brackets in stack = 2

We have to remove 2 brackets exactly to make brackets valid.

String → ( ) ( ) ) ) ( ( ) )

( ) ( ) ) ) ( ( ) ) → ( ) ( ) ( ( ) )

( ) ( ) ) ( ( ) ) → ( ( ) ) ( ( ) )

→ In last stack  
size is min count  
to remove brackets

String → ( ) ( ) { } [ ]<sup>8</sup>

Min Removal count = 2

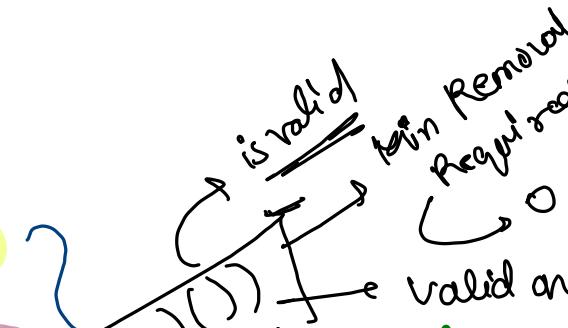
)<sub>8</sub>  
~~( )<sub>2</sub>~~  
){<sub>5</sub>  
~~{ }<sub>4</sub>~~  
~~[ ]<sub>2</sub>~~

( ) ( ) T ( ) T → ( ) ( ) ( )

( ) ( ) T ( ) ) → ( ) ( ) ( )

( ) ( ) ( ) T ) → ( ( ) ( ) )

( T ( ) ) ( ) [ ] → ( ( ) ( ) )



valid answer → print

if print → add it in Hashset

↓  
to avoid repetition

to avoid this  
Repetition, use  
Hashset<String>

