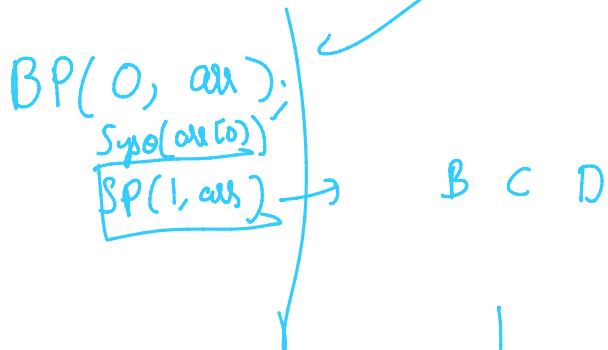


Array → Recursion

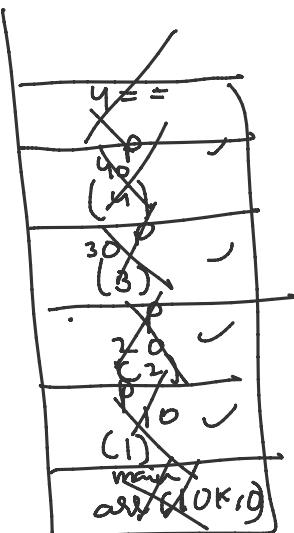
{ A, B, C, D }

A B, C, D



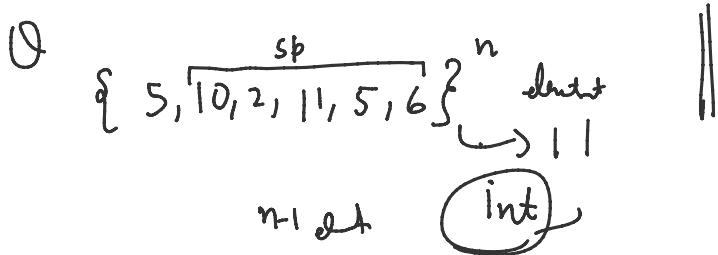
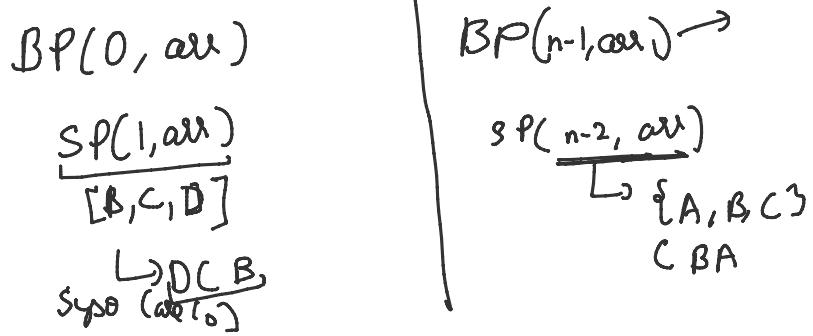
```
int[] arr = {10,20,30,40};
print(arr,0);
}
```

```
public static void print(int[] arr, int idx) {
    if(idx == arr.length) {
        return ;
    }
    System.out.println(arr[idx]);
    print(arr, idx+1);
}
```



Heap
10k
{ 10, 20, 30, 40 }

{ A, B, C, D } → D C B A

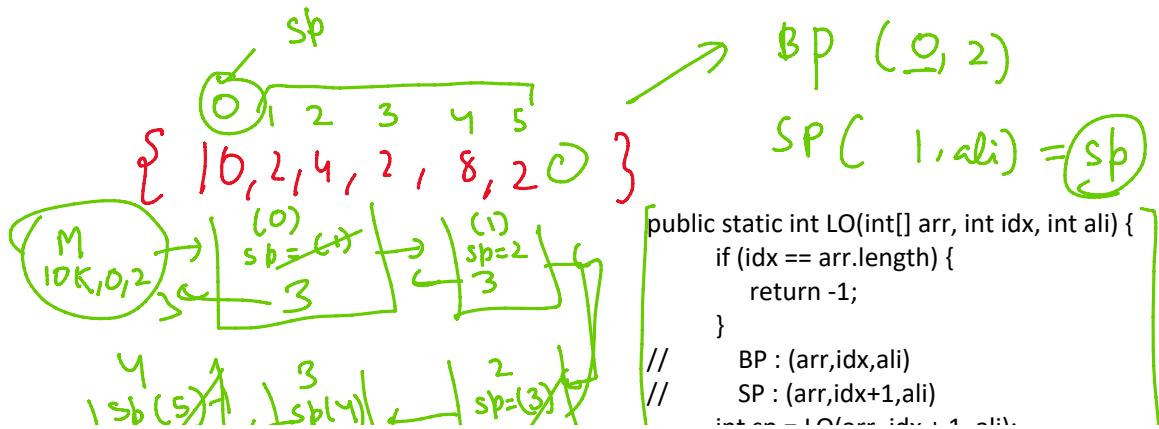
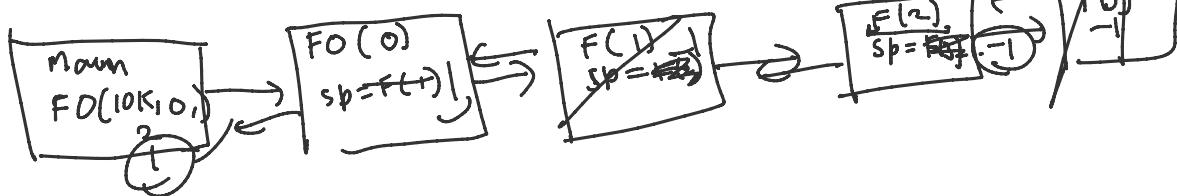


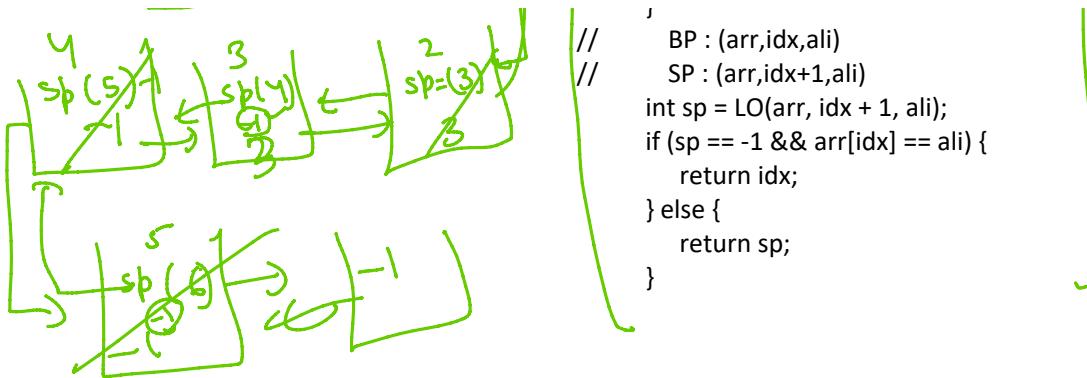
BP (0, all)
SP (1, all)

FO(2, arr)
 $\left\{ \begin{matrix} 10, & 2, & 4, \\ 0 & 1 & 2 \end{matrix} \right\}$

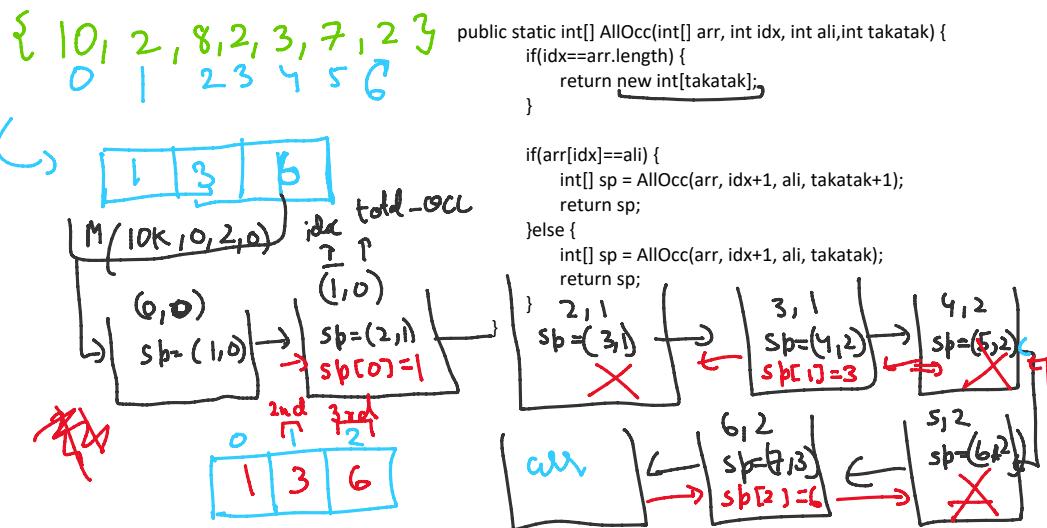
FO(2, arr)
BP
SP
 $(0, \text{all}, 2)$
 $(1, \text{all}, 2)$

```
public static int FO(int[] arr, int idx, int ali) {
    if(idx==arr.length) {
        return -1;
    }
    int sp = FO(arr, idx+1, ali);
    if(arr[idx]==ali) {
        return idx;
    } else {
        return sp;
    }
}
```





All occ



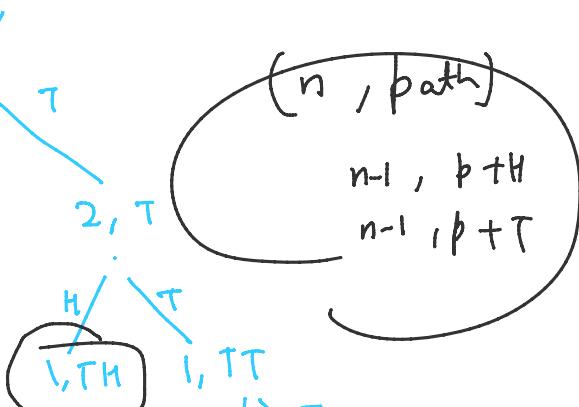
Coin Toss

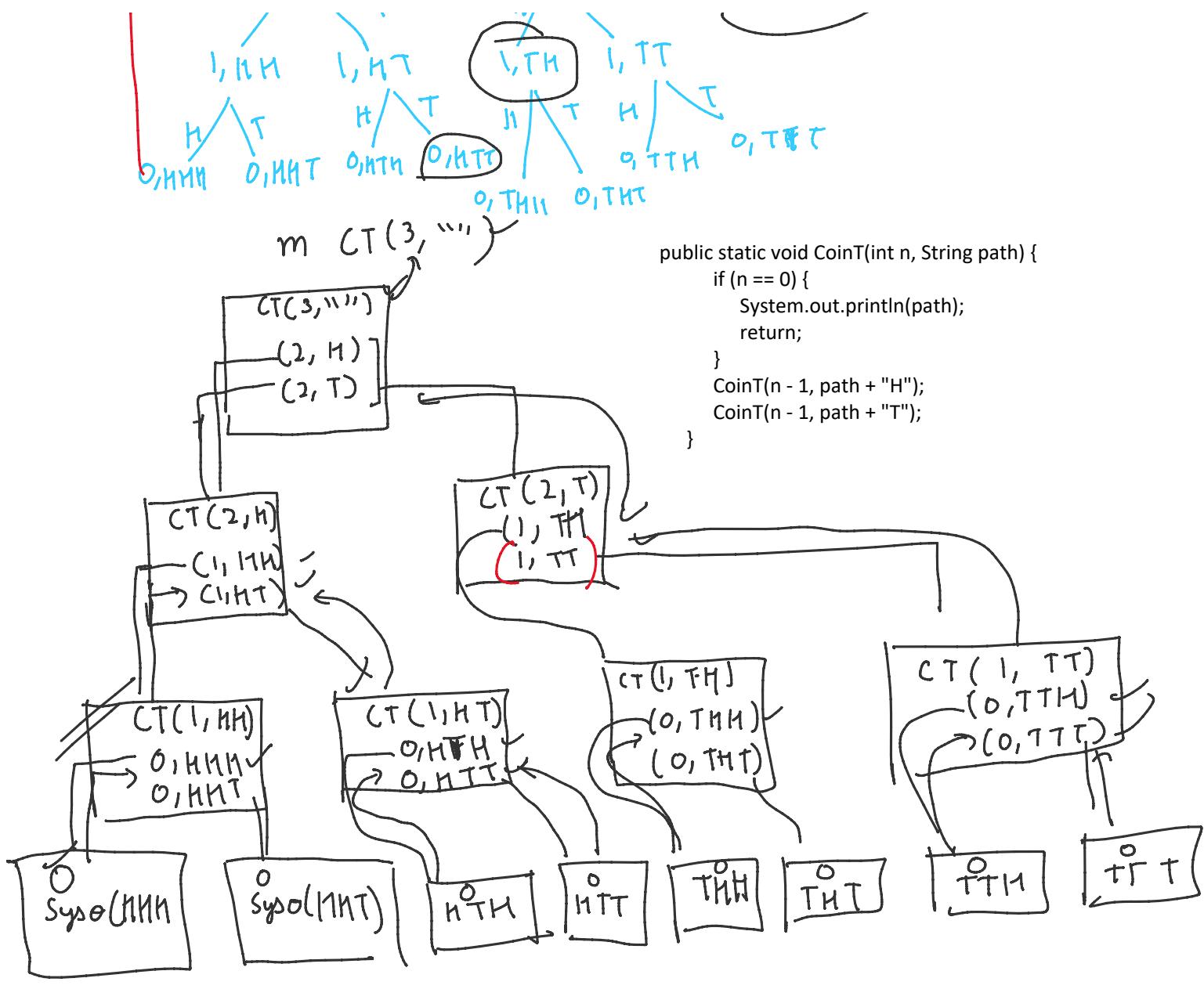
3
HHH
HHT
HTH
HTT

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

n

n-1

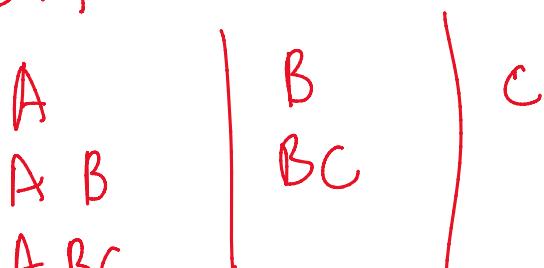




Subseq.

Subarray \rightarrow Subsequence & Cont

$$\{ A, B, C \dots \} \rightarrow \frac{n \cdot (n+1)}{2}$$

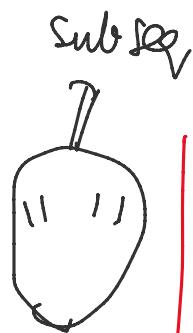


X D
A BC
~~Aj~~

||
||

Subseq → Sub, seq

{ A, B, C, D }



$\cup_{i=0}^n$
A
B
C
D

AB
AC
AD
BC
BD
CD

ABC
ABD
ACD
BCD

ABCD

$\cup_{i=0}^n$

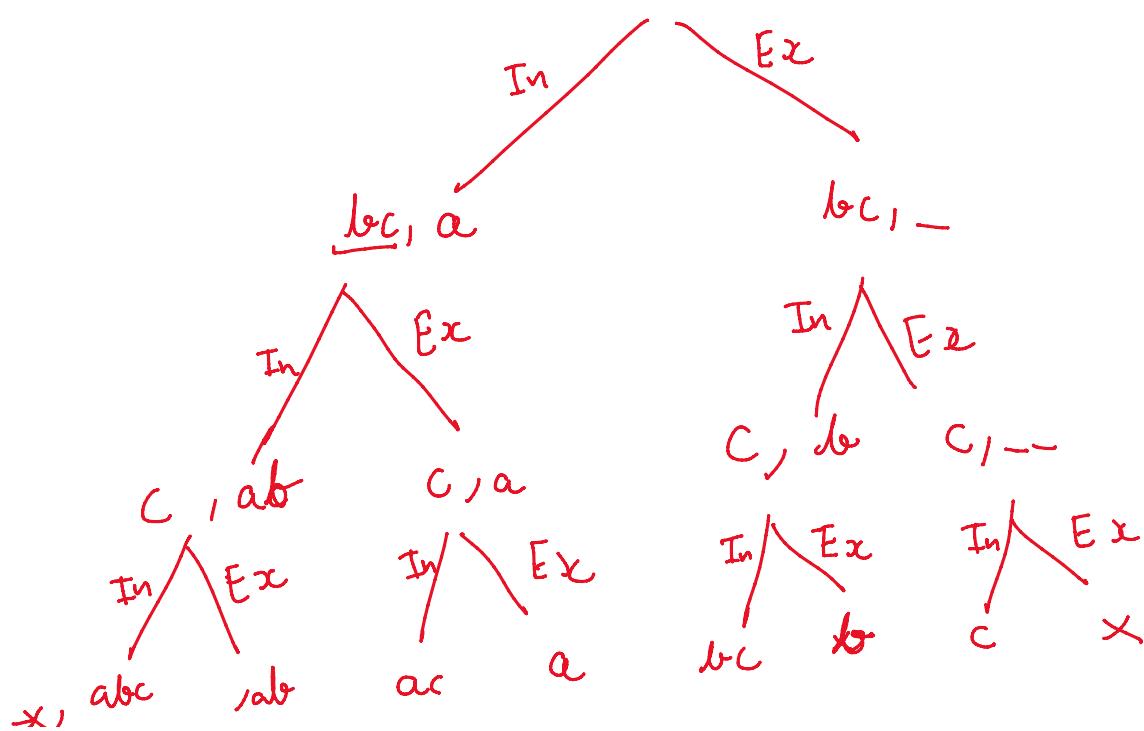
$$\sum_{r=0}^n {}^n C_r = 2^n$$

Subseq → Combinⁿ

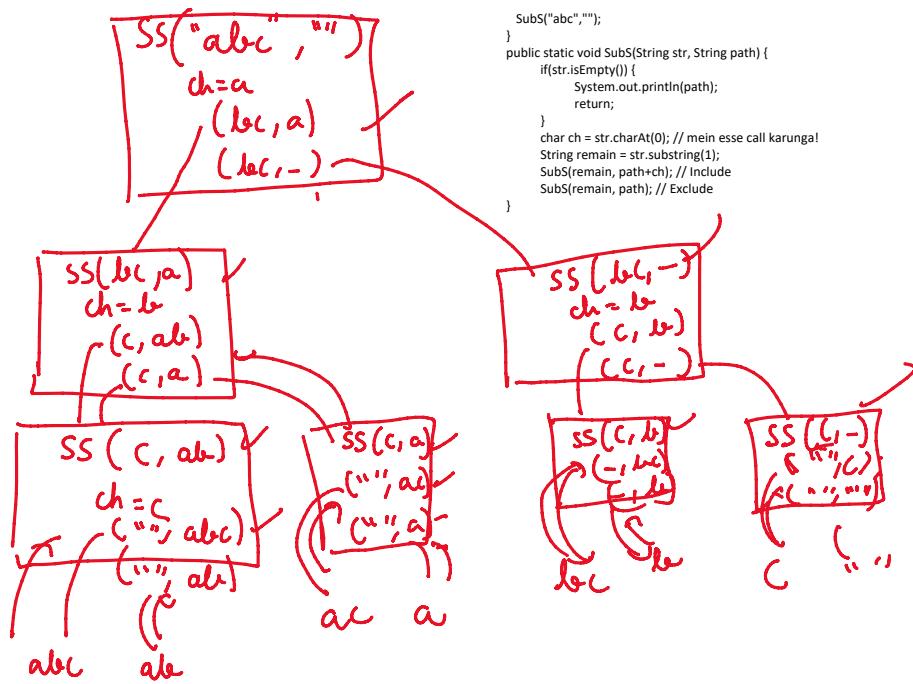
abc, "

— B — D

0 1 2
abc
|
bc
0 1

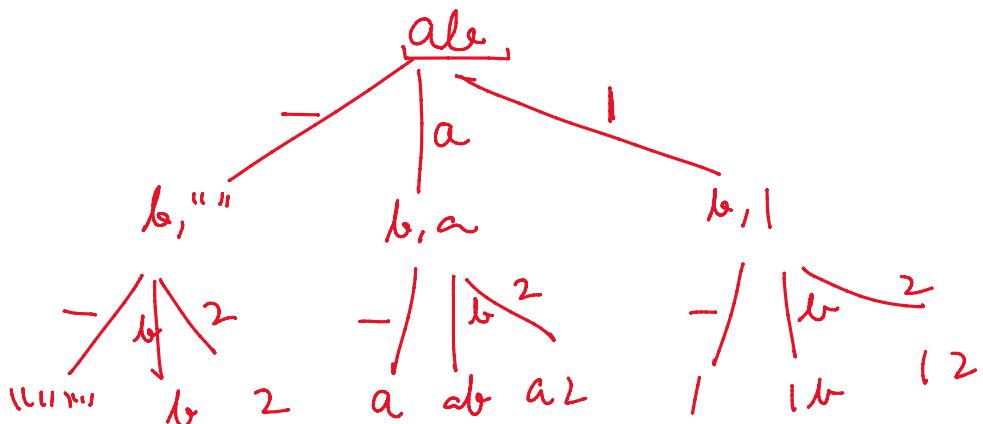
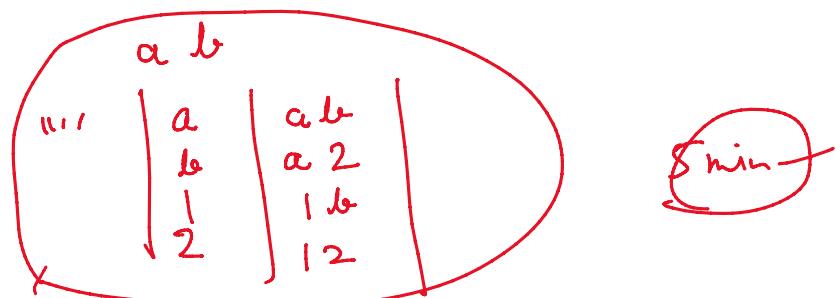


~~x~~ abc \abc ac a bc * c ~



Q Sub seq

ASCII



$\begin{matrix} 2, 3 \\ abc \end{matrix}$ $\begin{matrix} 4 \\ def \end{matrix}$



$\begin{matrix} adg \\ adh \\ adi \end{matrix}$

$\begin{matrix} aeg \\ aeh \\ aei \end{matrix}$

