① arr = [2, 2, 1, 0, 1, 2, 1, 0]

arr = [0, 0, 1, 1, 1, 2, 2, 2]

② start = 0

mid = 1

end = 2

pivot = 1

① start = 0, mid = 0, end = n-1;

② while (mid <= end)

{

if (a[mid] < pivot) ⇒ 0

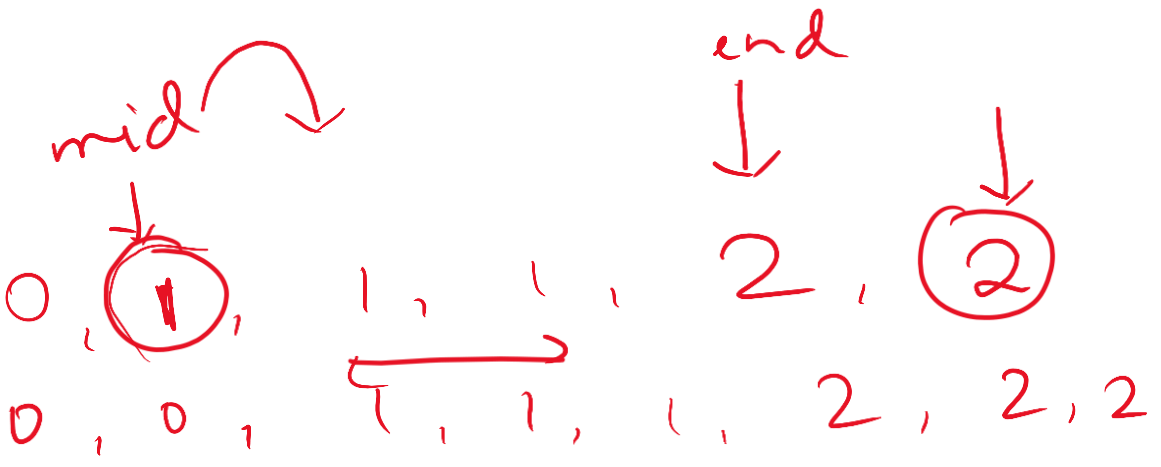
{ swap (A, start, mid);

start++;

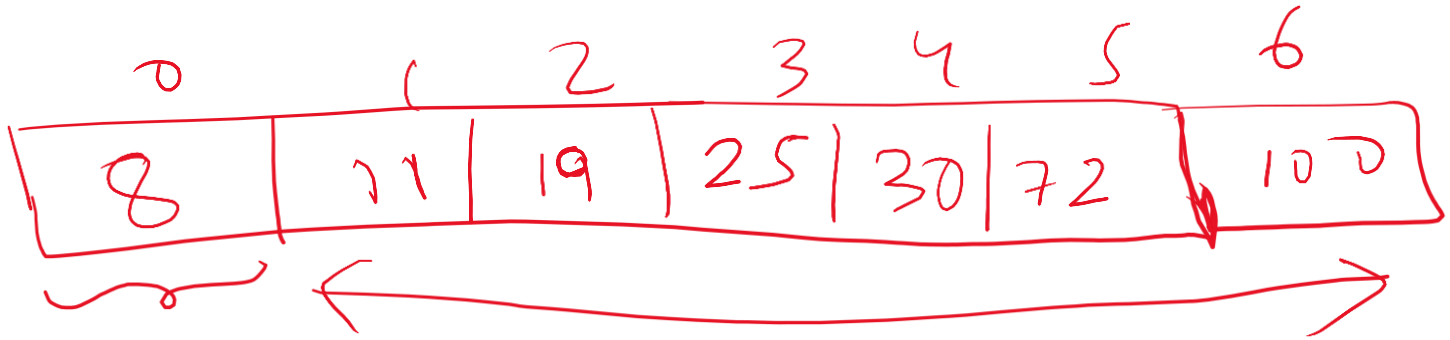
mid++;

}

```
else if ( a[mid] > pivot ) => 2
                1
{
    swap( A, mid, end );
    end--;
}
else
{
    mid++;
}
}
```



y =



y[1 --- 6] is sorted

first = y(0) = 30

for (int k = 1; k < n && y[k] < first; k++)

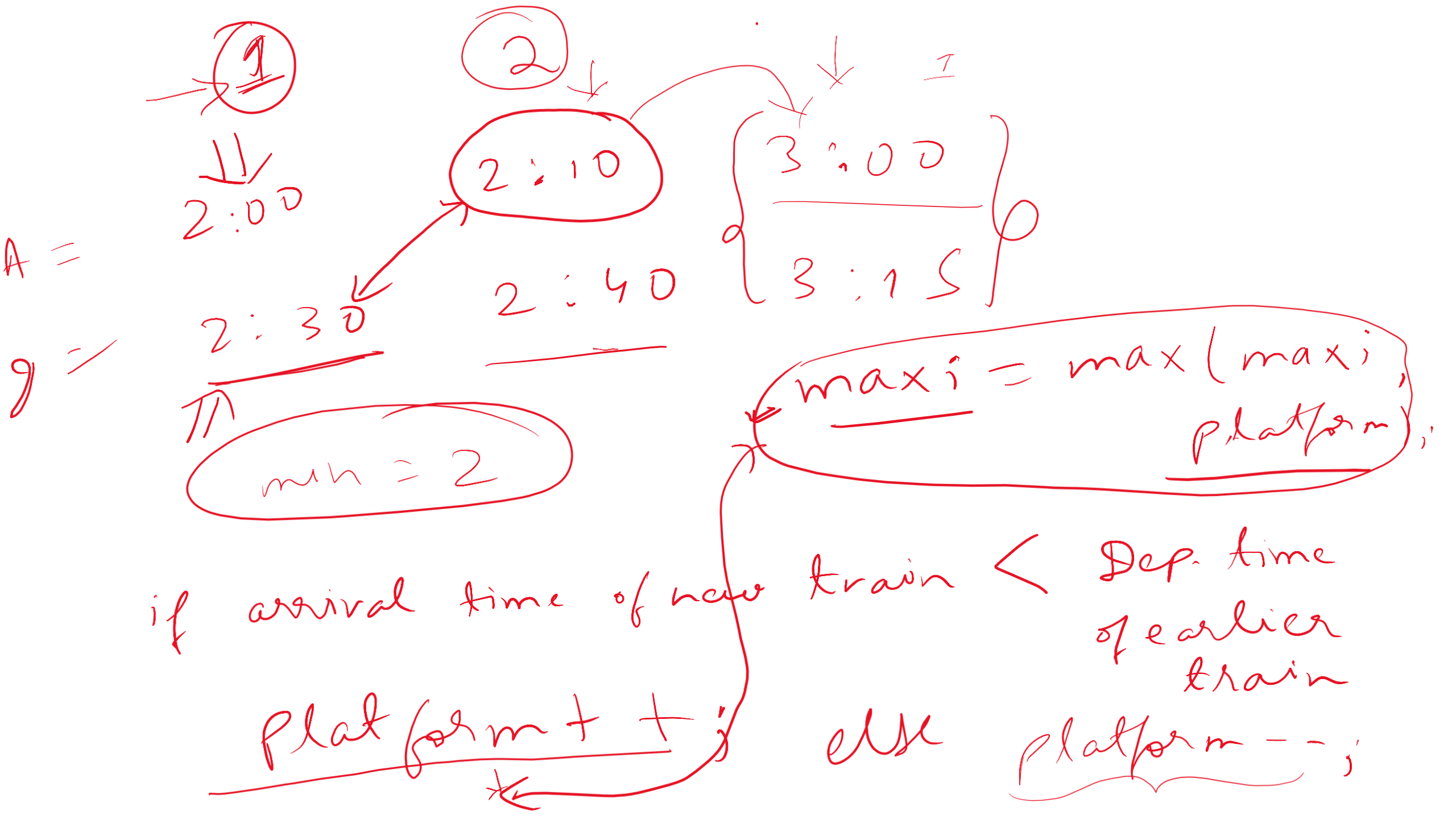
{ y[k-1] = y[k]; }

y[3] = y(4)

= 25

y[k-1] = first;

y[4] = 30


1
2
2:00
2:10
2:30
2:40
3:00
3:15
A =
D =
min = 2
maxi = max(maxi, platform);
if arrival time of new train < Dep. time of earlier train
platform++;
else
platform--;

T1 T2 T3 T4 T5 T6

arr = [2, 2.15, 2.30, 2.45, 3.15, 3.45]

dep = [2.18, 2.33, 2.48, 2.59, 3.59, 4:00]

tp

2

$P_1 = T5$

$P_2 = T6$

maxi = 2

count = 1

maxi = max(maxi, count) = 2

① -ve * -ve = +ve no

min_ending_till_here = a[0];

max_ending_till_here = a[0];

max_so_far = a[0];

```
for ( int i = 1 ; i < n ; i++ )
{
    int temp = max(a(i), a(i) * max_eh,
                   a(i) * min_eh);
    min_eh = min(a(i), a(i) * max_eh,
                 a(i) * min_eh);
    max_eh = temp;
    max_so_far = max(max_so_far, max_eh);
}
```

$x + y = \text{constant}$

$y = \text{constant} - x$

map [curr-sum]

key