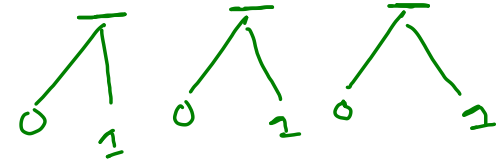


subsets

$ts = 8$

$[1, 2, 3]$

0	0	0	0	-	-	-
1	0	0	1	-	-	3
2	0	1	0	-	2	-
3	0	1	1	-	2	3
4	1	0	0	1	-	-
5	1	0	1	1	-	3
6	1	1	0	1	2	-
7	1	1	1	1	2	3



$$2 \times 2 \times 2 = 8$$

0 \rightarrow absent

1 \rightarrow present

```

public static void printSubsets(int[] arr) {
    int n = arr.length;
    int ts = (int) Math.pow(2, n);

    for (int d = 0; d < ts; d++) {
        int[] bin = decToBinary(d, n);

        //print subset
        for (int i = 0; i < n; i++) {
            if (bin[i] == 0) {
                //absent
                System.out.print("-\t");
            }
            else {
                //present
                System.out.print(arr[i] + "\t");
            }
        }

        System.out.println();
    }
}

public static int[] decToBinary(int dec, int len) {
    int[] ans = new int[len];
    int idx = ans.length - 1;

    while (dec > 0) {
        int rem = dec % 2;
        dec = dec / 2;

        ans[idx] = rem;
        idx--;
    }

    return ans;
}

```

10, 20, 30

$n = 3$

$ts = 8$

d

bin

subset

0



- - -

1



- - 30

2



- 20 -

3



- 20 30

4



10 - -

5



10 - 30

6



10 20 -

7



10 20 30

Binary Search

5	10	12	19	21	30	35	40	45	48	50	67
0	1	2	3	4	5	6	7	8	9	10	11

lo m hi

Search -> 35

```
int lo = 0
```

```
int hi = n-1;
```

```
while (lo <= hi) {
```

```
    int m = (lo + hi) / 2;
```

```
    if (arr[m] == data) {
```

```
        return m;
```

```
    }
```

```
    else if (arr[m] < data) {
```

```
        lo = mid + 1;
```

```
    }
```

```
    else {
```

```
        hi = mid - 1;
```

```
    }
```

```
}
```

$$\text{mid} = \frac{0 + 11}{2} = 5$$

$$\text{mid} = \frac{0 + 11}{2} = 8$$

$$\text{mid} = \frac{6 + 7}{2} = 6$$

5	10	15	17	25	30	40	50
0	1	2	3	4	5	6	7

$$\text{low} = 12$$

$$\text{hi} = 40$$

$$\text{low} = \cancel{10}$$

$$\text{hi} = \cancel{17} \quad \cancel{15}$$

$$\text{mid} = \frac{0 + 7}{2} = 3$$

$$\text{mid} = \frac{0 + 2}{2} = 1$$

$$\text{mid} = \frac{2 + 2}{2} = 2$$

First Index
Last Index

1	3	3	3	7	7	12	12	12	12	15
0	1	2	3	4	5	6	7	8	9	10

hi
lo
m

data = 12

li = ~~7~~ ~~8~~ 6

li = ~~7~~ ~~8~~ 9

```
while(lo <= hi) {
    int mid = (lo + hi)/2;

    if(arr[mid] == data) {
        li = mid;
        lo = mid+1;
    }
    else if(arr[mid] < data) {
        lo = mid+1;
    }
    else {
        hi = mid-1;
    }
}
```

```
while(lo <= hi) {
    int mid = (lo + hi)/2;

    if(arr[mid] == data) {
        fi = mid;
        hi = mid-1;
    }
    else if(arr[mid] < data) {
        lo = mid+1;
    }
    else {
        hi = mid-1;
    }
}

return fi;
```

max
sum
subarray

Kadane's algo

$[-2, 1, -3, 4, -1, 2, 1, -5, 4]$

-2	1	-3	4	-1	2	1	-5	4
0	1	2	3	4	5	6	7	8

— \rightarrow ms
— \rightarrow cs

cs = ~~1~~ ~~1~~ ~~-2~~ ~~4~~ ~~3~~ ~~5~~ ~~6~~ ~~1~~ 5

ms = ~~-2~~ ~~1~~ ~~4~~ ~~5~~ 6

```
int cs = arr[0];
int ms = arr[0];

for(int i = 1; i < arr.length; i++) {
    if(cs < 0) {
        //new start
        cs = arr[i];
    }
    else {
        //continue
        cs += arr[i];
    }

    if(cs > ms) {
        ms = cs;
    }
}

return ms;
```

— → msa

— → csa

-2	1	-3	4	-1	2	1	-5	4
0	1	2	3	4	5	6	7	8

$CS = \cancel{-2} \cancel{1} \cancel{-3} \cancel{4} \cancel{-1} \cancel{2} \cancel{1} \cancel{-5} \cancel{4}$

$ms = \cancel{-2} \cancel{1} \cancel{4} \cancel{-5} \cancel{6}$

$cst = \cancel{0} \cancel{1} \cancel{2} \cancel{3} \cancel{4} \cancel{5} \cancel{6} \cancel{7} \cancel{8}$

$cet = \cancel{0} \cancel{1} \cancel{2} \cancel{3} \cancel{4} \cancel{5} \cancel{6}$

$mst = \cancel{0} \cancel{1} \cancel{2} \cancel{3} \cancel{4}$

$mst = \cancel{0} \cancel{1} \cancel{2} \cancel{3} \cancel{4}$



```
for(int i = 1; i < arr.length; i++) {  
    if(cs < 0) {  
        //new start  
        cs = arr[i];  
        cst = i;  
        cet = i;  
    }  
    else {  
        //continue  
        cs += arr[i];  
        cet = i;  
    }  
  
    if(cs > ms) {  
        ms = cs;  
        mst = cst;  
        met = cet;  
    }  
}
```