Ones and Zeros

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				
	4				
	5				1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				
	4				
	5			1	1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				
	4				
	5		1		1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				
	4				1
	5		1	1	1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				
	4			1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3				1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3			1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				
	3		1	1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2				1
	3		1	1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2			1	1
	3		1	1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1				1
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1			1	1
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	1	1

cntZero=1, cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	1	1

```
cntZero=1, cntOne=1
new String[]{"10", "0001", "111001", "1", "0"}
```

```
str="0001"
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

str="0001"

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	2	2

cntZero=3, cntOne=1

new String[]{"10", "0001", "111001", "1", "0"}

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

str="0001"

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	2	2

cntZero=3, cntOne=1

new String[]{"10", "0001", "111001", "1", "0"}

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	1	1
	5		1	2	2

str="0001"

cntZero=3, cntOne=1

new String[]{"10", "0001", "111001", "1", "0"}

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

str="0001"

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	1	2
	5		1	2	2

cntZero=3, cntOne=1

new String[]{"10", "0001", "111001", "1", "0"}

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	2
	5		1	2	2

str="0001"

cntZero=3, cntOne=1

new String[]{"10", "0001", "111001", "1", "0"}

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

str="0001"

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	2
	5		1	2	2

cntZero=3, cntOne=1

new String[]{"10", "0001", "111001", "1", "0"}

str="111001"

i=5

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	2
	5		1	2	3

str="111001"

dp[i][j]=
dp[i-cntZero][j-cntOne]

cntZero=2, cntOne=4

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="1"

i=5

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	2
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	2
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	2
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="1"

i=4

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	1
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="1"

i=3

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	1	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	1
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="1"

i=2

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	1	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	1
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="1"

i=1

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	1	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="1"

i=0

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0				1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0			1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	3

str="1"
cntZero=0
cntOne=1

dp[i][j]=
dp[i-cntZero][j-cntOne]

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
str="0"
i=5
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	2	4

str="1"
cntZero=1
cntOne=0

dp[i][j]=
dp[i-cntZero][j-cntOne]+
1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		1	3	4

str="1"
cntZero=1
cntOne=0

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5		2	3	4

str="1"
cntZero=1
cntOne=0

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	2	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="0"

i=4

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

3==2+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		1	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

3==2+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4		2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	2
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="0"

i=3

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	2	3
	4		2	3	3
	5		2	3	4

str="1"
cntZero=1
cntOne=0

2<2+1=3

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		1	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

2<2+1=3

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3		2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

1<1+1=2

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	2
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

1<1+1=2

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

str="0"

i=2

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	2	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

2<2+1=3

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		1	3	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

2<2+1=3

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2		2	3	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

1<1+1=2

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1		1	2	2
	2	1	2	3	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

0<1+1=2

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
str="0"
i=1
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1	1	1	2	2
	2	1	2	3	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

2==1+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1	1	1	2	2
	2	1	2	3	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

2==1+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```

```
dp[i][j] = Math.max(dp[i][j], dp[i - cntZero][j - cntOne] +
1);
```

i/j					
		0	1	2	3
	0		1	1	1
	1	1	2	2	2
	2	1	2	3	3
	3	1	2	3	3
	4	1	2	3	3
	5	1	2	3	4

str="1"
cntZero=1
cntOne=0

1<1+1=2

dp[i][j]=
dp[i-cntZero][j-cntOne]+1

```
new String[]{"10", "0001", "111001", "1", "0"}
```