①

```
For (i=r; i<n; i++) {
    c=a+b;
    a=b;
    b=c; }
```

②

```
if (r>n-r) {
    r=n-r; }
for (i=0; i<r; i++) {
    result *= (n-i);
    result /= (i+1); }
return result;
```

③ برای نیترناجی درحالت بازگشتی پیچیدگی $O(n^2)$ است و درحالت غیر بازگشتی $O(n)$

برای سوال ② درحالت بازگشتی پیچیدگی $O(r)$ درحالت غیر بازگشتی نیز $O(r)$ است

④

```
func bigNumber(n):
    int bigNumber []   (.size = 300)  , size=1
    for i from 2 to n:
        m = 0
        for j from 0 to size-1:
            sth = bigNumber[j] * i + m
            bigNumber[j] = sth%10
            m = sth/10
            while m>0:
                bigNumber[size] = m%10
                m=m/10  , size=size+1
```

            return bigNumber

func   QuestionSix (n);   ④

        int   a array [] → size: nxn
        for i from o to n-1;
            for j from o to i ;
                if j==o or j==i ;
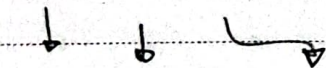   a [i][j] = 1
                else
                    a [i][j] = a [i-1][j-1] + a [i-1][j]
                print pascal [i][j] + print new line

                                                    سوال ② ⑨

m = ٠, ١, ٢
    ↓   ↓   ↓
١,٤١,٢٢ ١,٤١,٤٢ ١,٤١,٥٤ = ١١,٦ ⟶ O(s+m) در بدترین حالت داریم:

                                        به سادگی می‌توان گفت که

boolean is Transpose (int [][] a, int [][] b, int n) {   سوال③

    for (int i=o; i<n; i++) for (int j=o; j<n; j++)
    if (a[i][j] ! b[j][i])
    return false;
    else
    return true

                                        مثال ① سوال ③