

0 1 2 3 4 5 6
num 1 2 3 5 7 10 14

dfs(0,0)

result[0]=1

dfs(1,1)

result[1]=1

dfs(2,2)

result[2]=1

dfs(3,3)

result[3]=1

dfs(4,4)

result[4]=1

dfs(5,5)

result[5]=1

dfs(6,6)

print

result[6]=1

dfs(7,7) → X

result[6]=0

result[5]=0

result[6]=1

dfs(7,6)

print

result[6]=0

result[4]=0

result[5]=1

dfs(6,5)

result[6]=1

dfs(7,6)

print

result[6]=0

result[5]=0

result[6]=1

dfs(7,5)

result[6]=0

result[3]=0

result[4]=1

dfs(5,4)

result[5]=1

dfs(6,5)

result[6]=1

dfs(7,6)

print

result[6]=0

result[5]=0

result[6]=1

dfs(7,5)

result[6]=0

result[4]=0

1 2 3 5 7 10

1 2 3 5 7 14

1 2 3 5 10 14

1 2 3 7 10 14

1 1 1 1 | 0 0

1 1 1 1 | 0 0 0

1 1 1 1 | 0 1 0

1 1 1 1 | 0 1 1

1 1 1 1 | 0 1 0

1 1 1 1 | 0 0 0

1 1 1 1 | 0 0 1

1 1 1 1 | 0 0 0

1 1 1 0 | 0 0 0

1 1 1 0 | 1 0 0

1 1 1 0 | 1 1 0

1 1 1 0 | 1 1 1

1 1 1 0 | 1 1 0

1 1 1 0 | 1 0 0

1 1 1 0 | 1 0 1

1 1 1 0 | 1 0 0

1 1 1 0 0 0 0 0

```

result[5]=1      111 0010
dfs(6,4)
  result[6]=1    111 0011
  dfs(7,5)
    result[6]=0  111 0010
    result[5]=0  111 0000
    result[6]=1  111 0001
    dfs(7,4)
      result[6]=0  111 0000
      result[2]=0
      result[3]=1  1101000
      dfs(4,3)
        result[4]=1  1101100
        dfs(5,4)
          result[5]=1  1101110
          dfs(6,5)
            result[6]=1  1101111
            dfs(7,6)
              print
              result[6]=0
              result[5]=0
              dfs(6,4) result[6]=1 dfs(7,5) result[6]=0

```

1257 1014

이런식으로 반복. 갈수록 심숙없이
 부피가 커서 다 잡기 힘들...
 알고리즘 자체는 잘 모르겠지만 일단 이렇게 하면 모두 들수 있다는 건 알겠습니다...