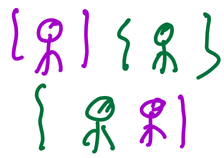
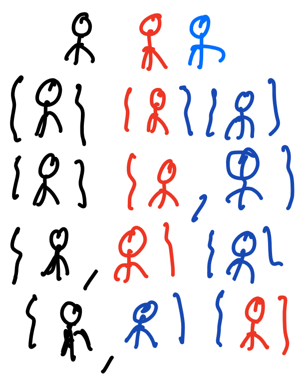
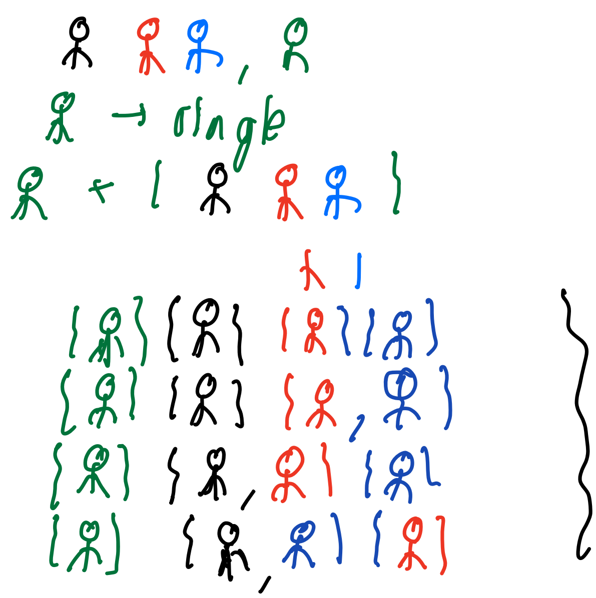
Q ⇒ Party pairs : Given N persons, how many ways we can pair all people.

Note: A person either wants to stay alone or get paired.

N = 1  → 1 way

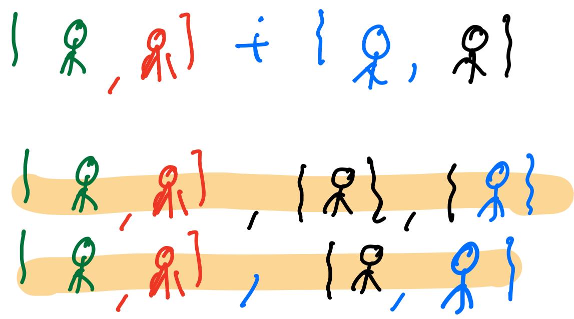
N = 2  → 2 ways

N = 3  → 4 way

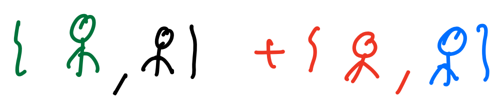
N = 4 

⇒ 4

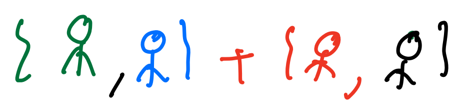
Pairs:



→ 2 pairs



→ 2 pairs



→ 2 pairs

10 pairs

$\{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$

single → {6 people}

pair → {5 people}

{6 people}

s → {5 people}

p → {4 people}

Party (5) = Party (4) + (4) * Party (3)

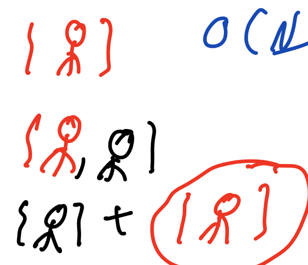
$dp[0] = 0$
$dp[1] = 1$

$dp[2] = dp[1] + 1 * dp[0]$
$= 1 + 1*0$
$= 1$

$\Rightarrow dp[0] = 1$
$dp[1] = 1$
$dp[2] = dp[1] + 1 * dp[0]$
$= 1 + 1*1$
$= 2$

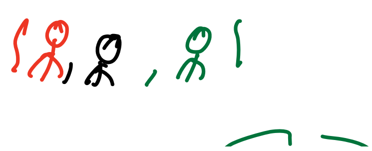
0	1	2	3	4
1	1	2	4	

$(N+1) * 1$
$O(N+2)$
$O(N)$



$dp[1] \Rightarrow$

$1 * dp[0] \Rightarrow$ { R, B } + 0



{ } + {R, B} ⇒ DP[2] = 2

{G, R} + DP[1] ⇒ 2*DP[1]

{G, B} + DP[1] ⇒ 2*1

{ , , , }

B + { DP[3]

```
dp[0]=1            dp[N+1]
dp[1]=1
for( i=2 ; i<=n; i++)
      dp[i]= dp[i-1] + (i-1) *dp[i-2];
 return dp[N];
}
```

TC: O(N)

SC: O(N)

↳ TODO (optimize)

Q ⇒ Min no. of perfect squares to be added to get Target sum.

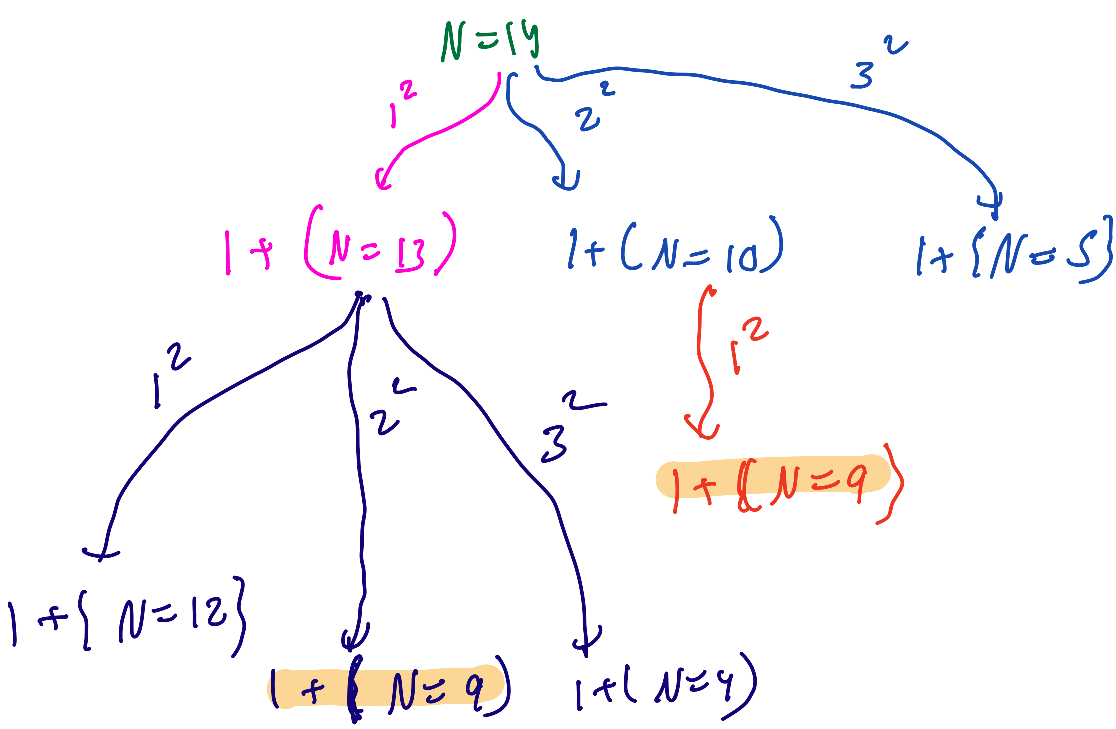
$N=6 \rightarrow 1^2 + 1^2 + 2^2 = 6 \Rightarrow 3$

$N=10 \rightarrow 1^2 + 3^2 = 10 \rightarrow 2$

$N=9 \rightarrow 3^2 = 9 \rightarrow 1$

$N=12 \rightarrow 3^2 + 1^2 + 1^2 + 1^2 \rightarrow 4$

$2^2 + 2^2 + 2^2 \rightarrow 3$



$$dp[i] = \min \begin{cases} dp[i - 1^2] + 1 \\ dp[i - 2^2] + 1 \\ \vdots \\ dp[i - j^2] + 1 \end{cases}$$

$j * j <= i$

$$dp[i] = \min \left[\forall_{j=1}^{j*j<=i} dp[i-j^2]+1 \right]$$

$1 = 1^2$

Base cond:

$dp[0] = 1$

$dp[1] = dp[1-1^2]+1$

$= dp[0]+1$

$= 1+1$

$= 2$

$dp[0] = 0$

$dp[1] = dp[1-1^2]+1$

$= dp[0]+1$

$= 0+1$

$= 1$

```
int func (int n)
{
    dp[N+1]
    dp[0] = 0
    for(i=1; i<=N; i++)
    {
        //ans = ____
        for( j=1; j*j<=i ; j++)
```

```
{
    ans = min( ans, dp[i-j*j] +1)
}
dp[i] = ans;
}
return dp[N];
}
```

$O(N) * O(\sqrt{N})$

\# states

$O(N * \sqrt{N})$

81

0	1	2	3	4
0	1	2	3	

$1^2 = 1$ $DP(1-1^2)+1$

$2 \Rightarrow 1^2+1^2=2$

$DP(2-1^2)+1$

$\Downarrow$

$DP(1)$

$3 \Rightarrow 1^2+1^2+1^2$

$\Rightarrow 3$

$DP(3 - 1^2) + 1$

$\Downarrow$

$DP(2) + 1$

$4 \Rightarrow 2^2 \Rightarrow 1$

4

$DP(4 - 1^2) + 1$ | $DP(4 - 2^2) + 1$

$DP(3) + 1$ | $DP(0) + 1$

$3 + 1$ | $0 + 1$

$\{1, 1, 1, 1\}$

$4 = 1^2 + 1^2 + 1^2 + 1^2 \Rightarrow 1^2 + 3 \Rightarrow 1^2 + DP(3)$

$4 = 2^2$

$$
\begin{tikzcd}
 & 5 \arrow[dl, "1^2"'] \arrow[d, "2^2"] \arrow[dr, "3^2"] & \\
DP(4) + 1 & DP(1) + 1 & \times
\end{tikzcd}
$$

$1 + 1$ | $\downarrow$

$\Rightarrow 2$ | $1 + 1 \Rightarrow 2$

$2^2 + 1^2$ | $2^2 + 1^2$

Q3 ⇒ Given N element find max subseq sum

Ex1: 2, -4, 5, 3, -8, 1 ⇒ 11

Ex2: 2, 6, -1, 4, 3, -5

Ex3: -4, -3, -8, -2

Ex4: 3, 2, 4, 8

Q4 ⇒ Given arr[N], find max subseq sum.

Note: In a subseq, 2 adjacent elements cannot be picked

Ex1: 9, 14, 3

Ex2: 9, 4, 13, 24

Ex3: 15, 14, 2

$$arr[8] = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2, & -1, & -4, & 5, & 3, & -1, & 4, & 2 \end{array}$$

dp[7]
a[0-7]

leave 7th index

pick it up

dp[6]
a[0-6]

dp[5]
a[0-5] + arr[7]

dp[5]
a[0-5]

a[0-4] + arr[6]

$$dp[i] = \max \begin{cases} arr[i] + dp[i-2] \\ dp[i-1] \end{cases}$$

Base Cond:

$i = 0$: $dp[0] = arr[0]$

$i = 1$: $dp[1] = \max(a[0], a[1])$

2, 4

```
int maxSub( int arr[] ),
  {
     n = arr.length;
     int dp[N];

   dp[0] = arr[0];
   dp[1] = max (arr[0], arr[1]);

   for( i = 2; i < n; i++)
     {
       dp[i] = max( dp[i-1], dp[i-2] + a[i],
                                      arr[i] )
     }

   return dp[n-1];
  }
```

O(N) ⇒ TC

O(N) ⇒ SC

↳ TODO

arr = $-2, -4, 1$

-2	-2	

$(-2, -2+1)$

$(-2, -1)$

(-1)

arr = $-2, -2, -1$

-2	-2	-1

$(-2, -2+(-1), -1)$

arr = $2, -1, -2$

2	2	2

$(2, 2+(-2), -2)$

$\Rightarrow 2$