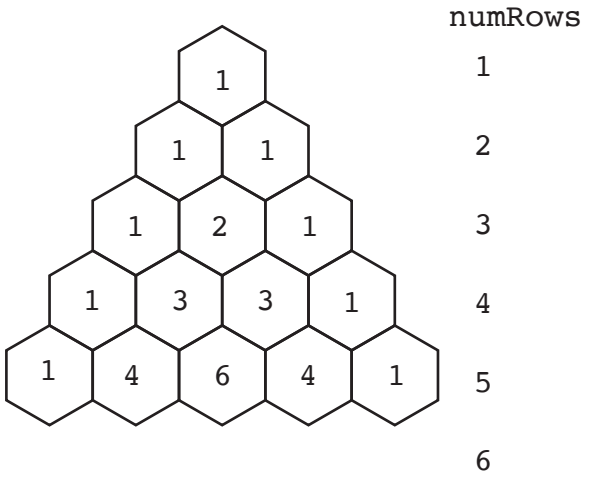
Pascal's triangle




```
def generate(numRows: int) -> List[List[int]]:
    rows = []

    if numRows <= 0:    # base-case
        return rows

    rows.append([1])

    for i in range(1, numRows):
        row = []
        for j in range(0, i + 1):
            if j == 0 or j == i:
                row.append(1)
            else:
                prev = rows[i - 1]
                row.append(prev[j - 1] + prev[j])
        rows.append(row)

    return rows
```

rows = []

rows = [[1]]

for loop

row = []

for loop

```
i = 1
  0 -> 2
  for j in range(0, 2):
    if j=0 or j==iv:
      row.append(1) -> row=[1]
    else:
      prev = rows[0]
      row.append(prev[i-1]+prev[j])
  rows.append(row)
```

row = [1,1]

rows = [[1]]
0

rows = [[1], [1,1]]
0 1

```
i = 2                    0 -> 1 -> 2 -> 3
  0 -> 3          row: [1,
  for j in range(0, 3):        [1,2
    if j=0 or j==i:                [1,2,1
      row.append(1) -> row=[1]
    else:
      prev = rows[1]    prev = rows[1] = [1,1]
      row.append(prev[i-1]+prev[j])
  rows.append(row)
```

row = [1,2,1]

rows = [[1], [1,1]]
0 1

rows = [[1], [1,1], [1,2,3]]
0 1 2

```
i = 3                    0 -> 1 -> 2 -> 3 -> 4
  0 -> 4          row: [1,
  for j in range(0, 4):        [1,3
    if j=0 or j==i:                [1,3,3
      row.append(1) -> row=[1]           [1,3,3,1
    else:                                  prev[j] = prev[1]
      prev = rows[1]    prev = rows[1] = [1,2,1]
      row.append(prev[i-1]+prev[j])              prev[i-1] = prev[2]
  rows.append(row)
```

row = [1,3,3,1]

rows = [[1], [1,1], [1,2,1]]
0 1 2

rows = [[1], [1,1], [1,2,1],[1,3,3,1]]
0 1 2 3

. . .