Friends Pairing Problem

```python
import collections


class Graph:
    def __init__(self):
        self.graph = collections.defaultdict(list)

    def addVertices(self, src, dest):
        self.graph[src].append(dest)
        self.graph[dest].append(src)
    def getGraph(self):
        return self.graph


class Solution:
    def getConnectedComponents(self, graph, n):
        visited = [False] * n
        comps = []
        for i in range(n):
            if visited[i] is False:
                comp = []
                self.getConnectedComponentsHelper(graph, i, visited, comp)
                comps.append(comp)
        return comps

    def getConnectedComponentsHelper(self, graph, src, visited, comp):
        visited[src] = True
        comp.append(src)

        for nbr in graph[src]:
            if visited[nbr] == False:
                self.getConnectedComponentsHelper(graph, nbr, visited, comp)


graph = Graph()
graph.addVertices(0, 1)
graph.addVertices(2, 3)
graph.addVertices(4, 5)
graph.addVertices(4, 6)
```

```
graph.addVertices(5, 6)
grp = graph.getGraph()
solution = Solution()
comps = solution.getConnectedComponents(grp, 7)
ans = 0
for i in range(len(comps)-1):
    for j in range(i+1,len(comps)):
        ans+=len(comps[i])*len(comps[j])
print(ans)
```