

# CS7200: Algorithm Design and Analysis

## Assignment-1

Sai Vijay Kumar Surineela

U01096171

Surineela.2@wright.edu

Manoj Kumar Reddy Avula

U01067535

avula.25@wright.edu

Prabhu Charan Murari

U01099304

murari.4@wright.edu

**Pseudo code: -**

**assignmet1.py:**

'''

Assume 'G1 & G2' as genders.

First create an array of size  $n \times n$  for the given input, with first element being the Gender in question and its other gender preferences.

Then create a dict 'prefGen' with gender in question as a key and its preferences as a list in the value.

Create two free lists, each list contains all the members of the similar gender.

Create a dict 'DG1' with each member of one gender as keys and its stable partners as value, these stable partner will be filled by the end.

Create another dict 'DG' with all the members and its partners.

Initialise a proposal count variable count with 0

while there are elements in G1:

    for each member A in G1:

        for each member B in the preference list of A:

            if B in free list of G2:

                remove B from free list of G2

                remove A from free list of G1

                Add B as value to A in DG1

                Add A as value to B in DG

                Add B as value to A in DG

                count = count + 1

                break from the loop and go to member of free list G1

```

else: //this means that B is free it is paired
    fetch the current partner of B from DG
    fetch the index of cuurent partner from the prefGen[B]
    fetch the index of A from the prefGen[B]
    if index of current partner is less than index of A:
        count = count + 1
        continue to next pref of A
    else:
        add current partner to free list of G1
        remove A from free list of G1
        update B as value to A in DG1
        add A as value to B in DG
        update B as value to A in DG
        count = count + 1
        break from the loop and go to member of free list G1

```

write the Dict DG1 to output.txt as follow:

```

A1 B1
A2 B2
A3 B3
'''

```

### **stabilityChecker.py:**

```
'''
```

Assume 'G1 & G2' as genders.

First create an array of size  $n*n$  for the given Input.txt, with first element being the Gender in question and its other gender preferences.

Then create a dict 'prefGen' with gender in question as a key and its preferences as a list in the value.

Create an array 'output' of size  $n*2$  for the given OutputToBeVerified.txt, as[[A1, B1],[A2, B2]....].

Create a List LG2 with all the members of G2 from slicing the list output.

for i in range(n):

```
    member A of G1 is output[i][0]
```

```

member B of G2 is output[i][1]
preferenceA is preference list of A from prefGen[A]
for B' from the list preferenceA:
    if B' == B: // this means that B is perfect match for A, also the loop will not go below the member of
B
        break out of the loop and go to next Pair.
    find index ind of B' from LG2, also the index is same for partner of B' in output
    find the current partner of B', A' from the index above using output[ind][0]
    preferenceB from prefGen[B]
    rankA' is the rank of A' from preferenceB
    rankA is the rank of A from preferenceB
    if rankA' is more than rankA:
        return False which means unstable

return True, which means stable

print the result stable or not to verified.txt

'''

```

**Note: same Pseudo code is provided in the code itself.**

**Time complexity:** The time complexity of the code is  $O(n^2)$ .