Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Дискретна математика

Лабораторна робота №2

«Бінарні відношення та їх основні властивості, операції над відношеннями»

Виконав:

студент групи ІО-62

Бурбіль М.А

Номер залікової книжки № 6203

Перевірив Новотарський М. А.

Київ

2017 р.

**Тема:** «Бінарні відношення та їх основні властивості, операції над відношеннями»

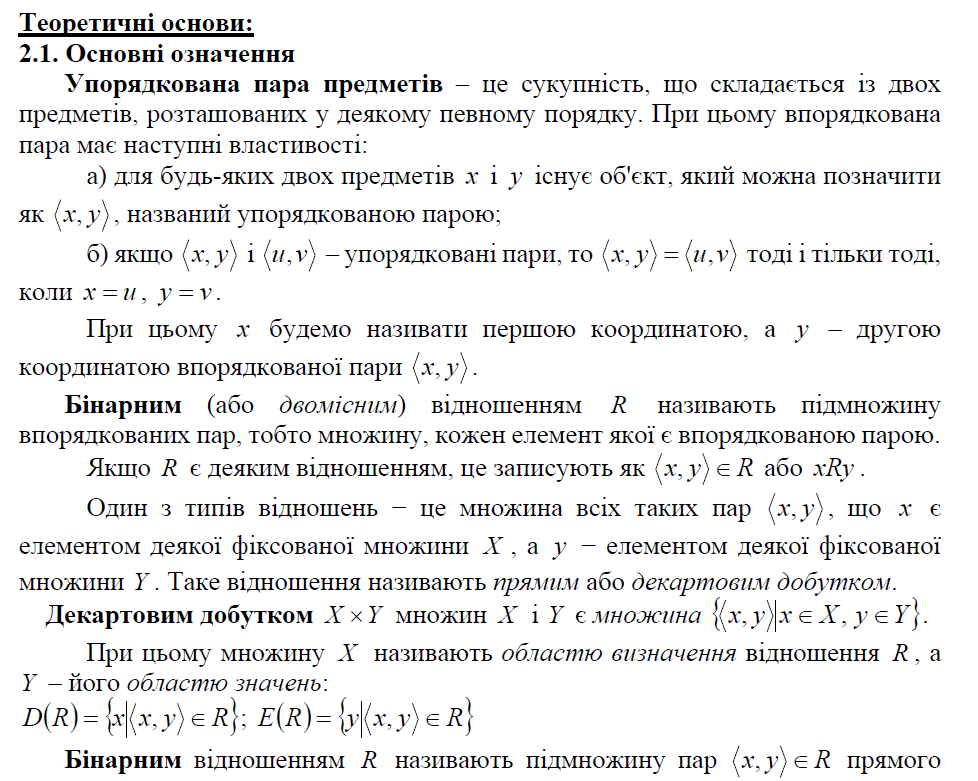
**Мета:** вивчити основні властивості бінарних відношень та оволодіти операціями над бінарними відношеннями.

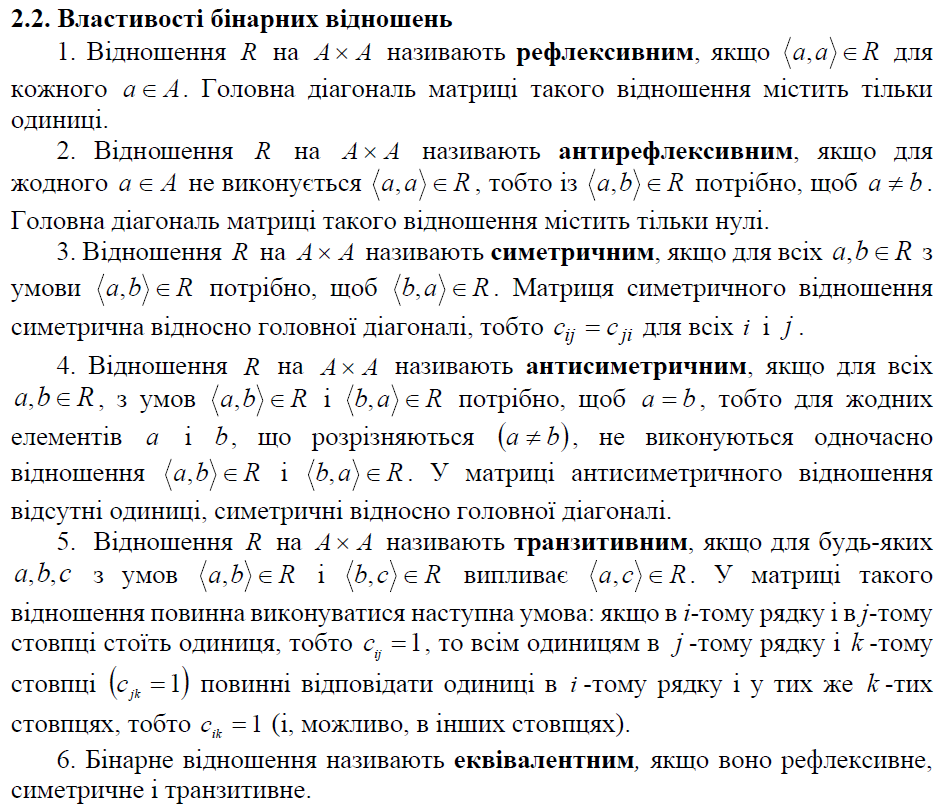
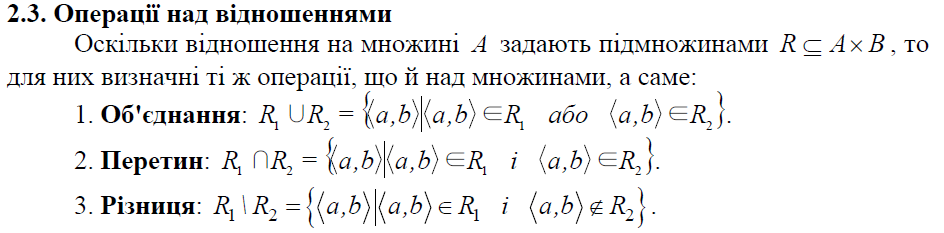
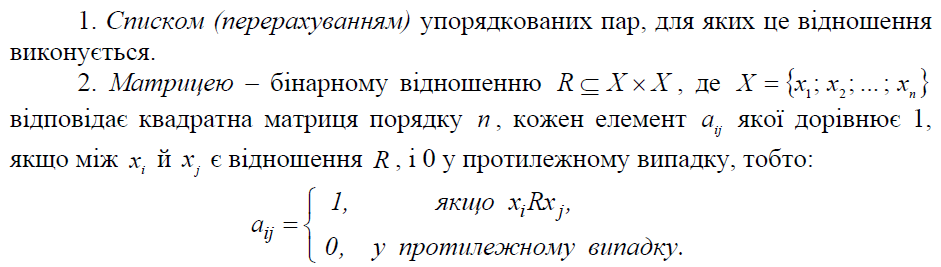
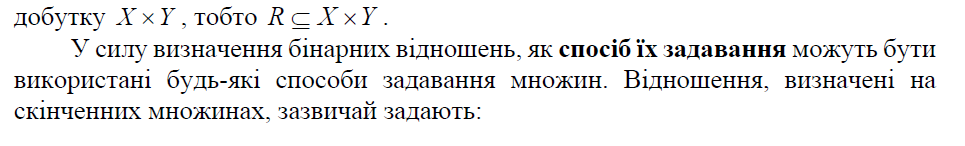
**Загальне завдання:**

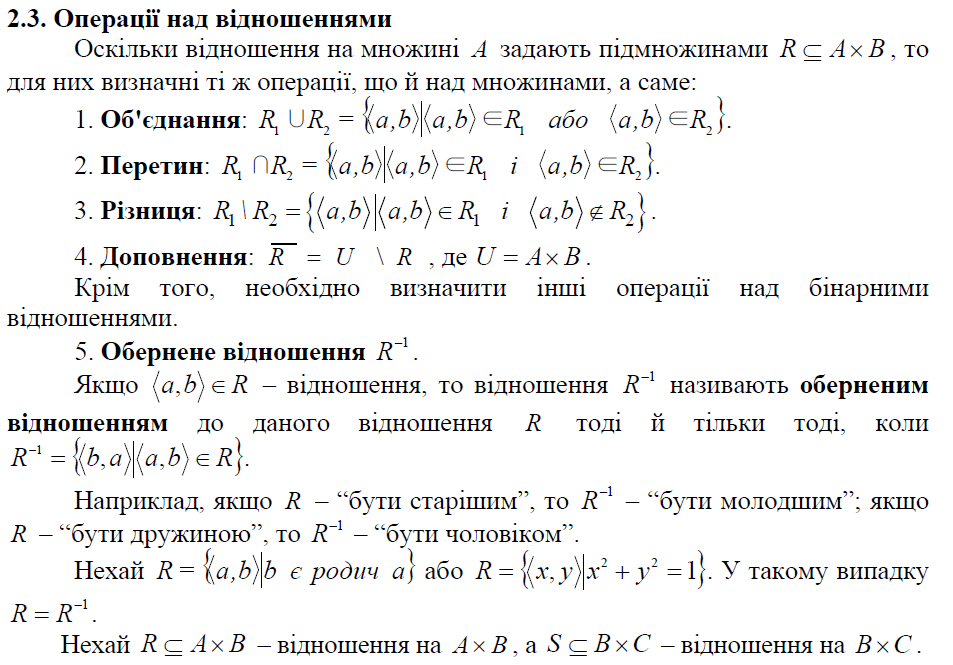
1. Написати в окремому модулі функцію для формування несуперечливих бінарних відношень.

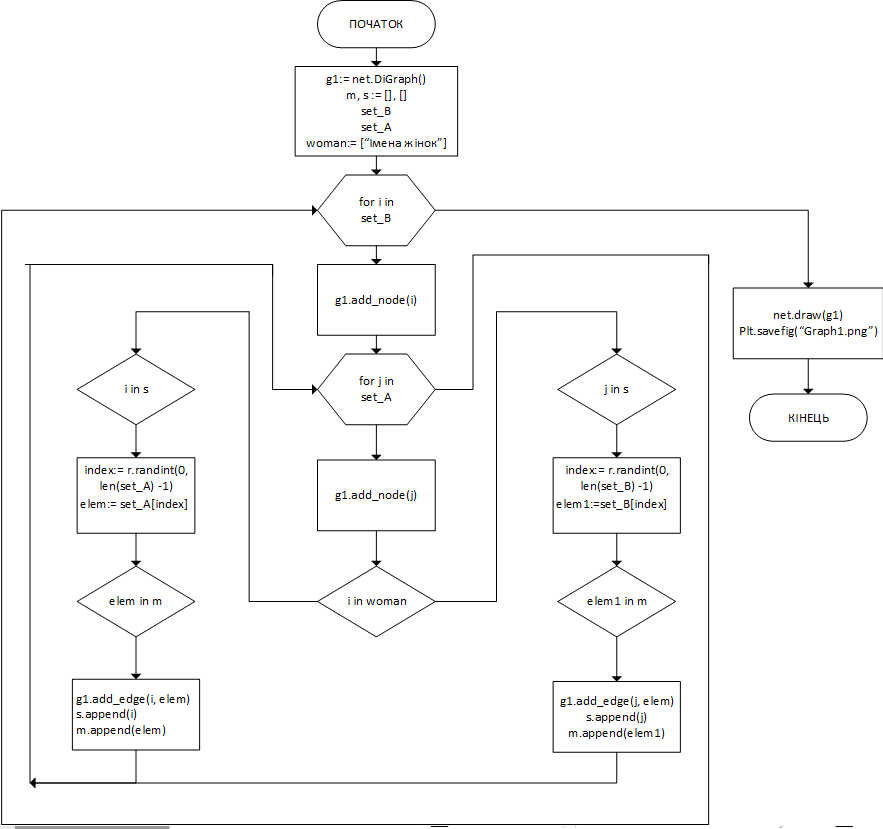
2. Написати в окремому модулі функції виконання логічних операцій над бінарними відношеннями.

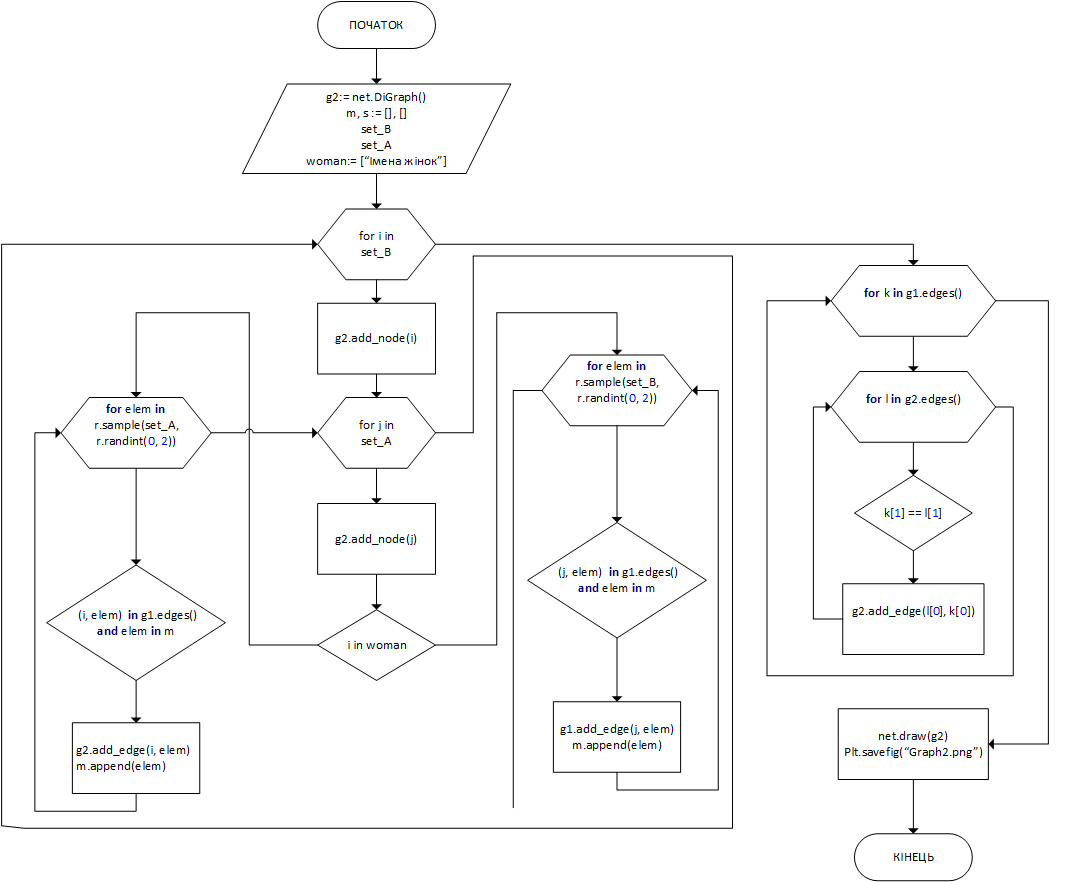
3. Пояснити правило формування несуперечливих відношень відповідно до Вашого варіанту.

**Індивідуальне завдання (Варіант №6)**

********



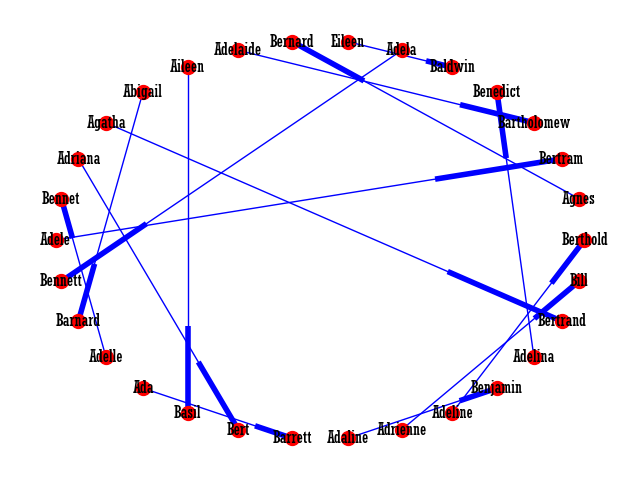
**Блок-схема для першого відношення.**

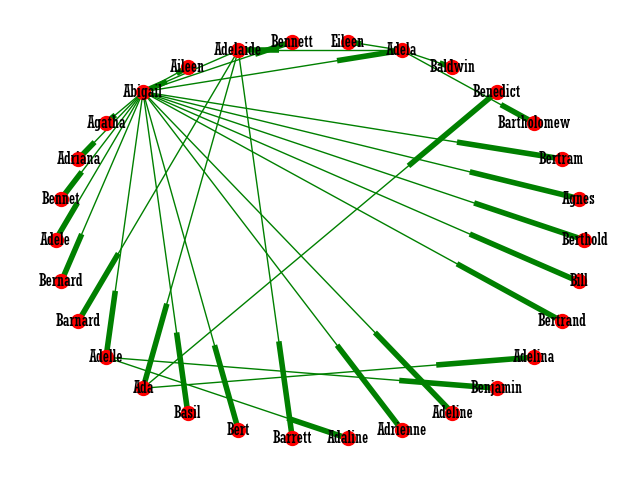
**Блок-схема для другого відношення.**

**Код програми, написаний власноруч.**

**import** random **as** r  
**import** pylab **as** plt  
**import** networkx **as** net  
  
**class** Graph1:  
 **def** \_\_init\_\_(self):  
 **with** open(**r"Set\_A.txt"**, **"r"**, encoding=**"UTF-8"**) **as** f:  
 set\_A = f.read()  
 **with** open(**r"Set\_B.txt"**, **"r"**, encoding=**"UTF-8"**) **as** f:  
 set\_B = f.read()  
 set\_A = set\_A.rstrip(**","**).split(**","**)  
 set\_B = set\_B.rstrip(**","**).split(**","**)  
 **with** open(**r"woman.txt"**, **"r"**, encoding=**"UTF-8"**) **as** f:  
 woman = f.read()  
 woman = woman.rstrip(**","**).split(**","**)  
  
  
 plt.figure(1)  
  
 self.g1 = net.DiGraph()  
 m, s = [], []  
 **for** i **in** set\_B:  
 self.g1.add\_node(i)  
 **for** j **in** set\_A:  
 self.g1.add\_node(j)  
 **if** i **in** woman:  
 index = r.randint(0, len(set\_A) - 1)  
 elem = set\_A[index]  
 **if** i **not in** s **and** elem **not in** m:  
 self.g1.add\_edge(i, elem, weight=15)  
 s.append(i)  
 m.append(elem)  
 **else**:  
 index = r.randint(0, len(set\_B) - 1)  
 elem1 = set\_B[index]  
 **if** j **not in** s **and** elem1 **not in** m:  
 self.g1.add\_edge(j, elem1, weight=15)  
 s.append(j)  
 m.append(elem1)  
 edges = self.g1.edges()  
  
 colors =[**'b' for** i **in** range(0, len(edges))]  
 net.draw(self.g1, pos=net.shell\_layout(self.g1), arrows=**True**, with\_labels=**True**, node\_size=100, width=1,  
 font\_size=13, font\_family=**"Rockwell Condensed"**, edge\_color=colors)  
  
 plt.savefig(**r"Graph1.png"**)  
  
 *# Second Graph* plt.figure(2)  
 self.g2 = net.DiGraph()  
 m = []  
 **for** i **in** set\_B:  
 self.g2.add\_node(i)  
 **for** j **in** set\_A:  
 self.g2.add\_node(j)  
 **if** i **not in** woman:  
 **for** elem **in** r.sample(set\_B, r.randint(0, 2)):  
 **if** (j, elem) **not in** self.g1.edges() **and** elem **not in** m:  
 self.g2.add\_edge(j, elem, weight=15)  
 m.append(elem)  
  
 **else**:  
 **for** elem **in** r.sample(set\_A, r.randint(0, 2)):  
 **if** (i, elem) **not in** self.g1.edges() **and** elem **not in** m:  
 self.g2.add\_edge(i, elem, weight=15)  
 m.append(elem)  
  
 **for** k **in** self.g1.edges():  
 **for** l **in** self.g2.edges():  
 **if** k[1] == l[1]:  
 **if** (l[0], k[0]) **not in** self.g2.edges() **or** (k[0], l[0]) **not in** self.g2.edges():  
 self.g2.add\_edge(l[0], k[0], weight=15)  
  
 edge = self.g2.edges()  
 colors =[**'g' for** i **in** range(0, len(edge))]  
 net.draw(self.g2, pos=net.shell\_layout(self.g2), arrows=**True**, with\_labels=**True**, node\_size=100, width=1,  
 font\_size=13, font\_family=**"Rockwell Condensed"**, edge\_color=colors)  
  
 plt.savefig(**r"Graph2.png"**)

**Результат програми**

****



**Висновок:** під час виконання лабораторної роботи найчастіше виникали помилки з роботою модуля tkinter. Проблеми усувалися шляхом вивчення нових можливостей цієї бібліотеки. Аналіз результатів показав, що власний алгоритм працює дещо швидше, ніж вбудовані функції Python.