

Speciális algoritmusok és adatstruktúrák

Beadandó feladatok

I. FELADAT

Adott egy szöveges fájl, amelynek sorai kémiai molekulákat írnak le SMILES formátumban. Írjunk programot, amely egy ilyen inputfájlt beolvas, majd egy szöveges kimeneti fájlba kiírja a szerkezetileg ekvivalens (izomorf) molekulapárok sorszámainak listáját.

Specifikáció:

Készítsünk el egy konzolos programot, amely két parancssori paramétert kap: az input és az output fájl nevét (ebben a sorrendben). A bemenet egy szöveges fájl, amelynek minden sora egy-egy molekulagráfot ír le SMILES karaktersorozat formájában. A fájlt egy üres sor zárja, vagyis az utolsó SMILES kód után is van egy sorvége jel.

A kimenet szintén egy szöveges fájl legyen, amelynek első sora egy k szám, az inputban található izomorf molekulagráf-párok sorszáma. A további k sora pedig ezen párokat adja meg $i j$ formában, ahol i és j a két molekula sorszáma (1-től kezdjük a számozást), és $i < j$. A párokat lexikografikusan növekvő sorrendben kell kiírni (i szerint, azon belül j szerint növekvően).

Megjegyzés: A matematikai értelemben különböző, de kémiai jelentése szerint ekvivalens molekulagráfokat különbözőnek tekintjük! Pl. az OC1=CC=CC=C1O és OC1=C(O)C=CC=C1 SMILES kóddal megadott molekulákat nem tekintjük izomorfoknak (az egyszeres és kétszeres kötéseket nem cseréljük le aromás kötésekre).

Példák:

INPUT:

```
C1=C(C1)C=CC=C1NC(=O)CC
C1=C(C=CC=C1C1)NC(=O)CC
C1=CC=C(C1)C=C1NC(CC)=O
C(CC)(=O)NC1=C(C=CC=C1)C1
C1(=C(C=CC=C1)NC(CC)=O)C1
C(NC1=C(C=CC=C1)C1)(CC)=O
O=C(NC1=CC=C(C=C1)C1)CC
CCC(NC1=CC=C(C1)C=C1)=O
CCC(=O)NC1=CC=CC(C1)=C1
C1C1=CC=C(C=C1)NC(=O)CC
```

OUTPUT:

```
8
1 9
2 3
4 5
4 6
5 6
7 8
7 10
8 10
```

INPUT:

```
CC1=C2C(C(=C(C=N2)C(OCC)=O)O)=CC=C1C1
O=C(OC)C1=C(C2=C(C(C)=C(C=C2)C1)N=C1)O
C1(=C(O)C2=C(C(C)=C(C=C2)C1)N=C1)C(OCC)=O
C1=NC2=C(C(=CC=C2C(O)=C1C(=O)OCC)C1)C
C1(=C(C=CC2=C1N=CC(C(OCC)=O)=C2O)C1)C
C12=C(C(=C(C1)C=C1)N=CC(=C2O)C(=O)OC
O=C(OCC)C1=C(C2=C(N=C1)C(=C(C=C2)C1)C)O
C(C1=C(O)C2=C(N=C1)C(C)=C(C=C2)C1)(=O)OC
```

OUTPUT:

```
7
1 4
2 6
2 8
3 5
3 7
5 7
6 8
```

II. FELADAT

Adott egy szöveges fájl, amelynek sorai kémiai molekulákat írnak le SMILES formátumban. Írjunk programot, amely egy ilyen inputfájlt beolvas, majd egy szöveges kimeneti fájlba kiírja azon molekulapárok sorszámait, ahol az első molekula részstruktúrája a másodiknak. Továbbá megadja azt is, hogy hányszor található meg az első struktúra a másodikban, vagyis hány részgráf-izomorfizmus függvény található közöttük.

Specifikáció:

Készítsünk el egy konzolos programot, amely két parancssori paramétert kap: az input és az output fájl nevét (ebben a sorrendben). A bemenet egy szöveges fájl, amelynek minden sora egy-egy molekulagráfot ír le SMILES karaktersorozat formájában. A fájlt egy üres sor zárja, vagyis az utolsó SMILES kód után is van egy sorvége jel.

A kimenet szintén egy szöveges fájl legyen, amelynek első sora egy k szám, amely megadja azon molekulapárok számát, ahol az első részstruktúrája a másodiknak. A kimenet további k sora pedig ezen párokat adja meg $i j c$ formában, ahol i és j a két molekula sorszáma (1-től kezdjük a számozást), c pedig az i . molekulát a j . molekulára leképező részgráf-izomorfizmus függvények száma. Ezeket a sorokat lexikografikusan növekvő sorrendben kell kiírni (i szerint, azon belül j szerint növekvően). Fontos, hogy $i=j$ eset is szerepeljen a kimenetben.

Megjegyzés: A matematikai értelemben különböző, de kémiai jelentése szerint ekvivalens molekulagráfokat különbözőnek tekintjük! Pl. az OC1=CC=CC=C1O és OC1=C(O)C=CC=C1 SMILES kóddal megadott molekulákat nem tekintjük izomorfoknak (az egyszeres és kétszeres kötéseket nem cseréljük le aromás kötésekre).

Példák:

INPUT:

```
CCC (=O) NC1=CC=CC (N) =C1
CNC1=CC=CC (N) =C1
NC1=CC=CC=C1
C1=CC=CC=C1
CN
```

OUTPUT:

```
14
1 1 1
2 1 1
2 2 1
3 1 2
3 2 2
3 3 1
4 1 6
4 2 6
4 3 6
4 4 6
5 1 3
5 2 3
5 3 1
5 5 1
```

INPUT:

```
CCC (=O) NC1=CC=CC=C1C1
CC (=O) NC1=CC (C1) =CC=C1
C1C1=CC=CC=C1
NC1=CC (C1) =CC=C1
CCC (=O) NC
CCC (N) =O . C1=CC=CC=C1
```

OUTPUT:

```
12
1 1 1
2 2 1
3 1 1
3 2 1
3 3 1
3 4 1
4 2 1
4 4 1
5 1 1
5 5 1
6 1 6
6 6 6
```