

Wstęp do programowania 2014

Pracownia 12

Uwaga: Informacja o wprawkach pojawi się na KNO. Podczas tych zajęć można oddawać wszystkie zadania z listy 11 za 100% punktów. Wszystkie zadania z tej listy będą ważne do końca semestru. Do maksimum wliczają się 3 punkty i **wartość maksimum już nie będzie modyfikowana**.

Premia za tę listę wynosi 0.5, przyznawana jest osobom, które zdobyły co najmniej 2p za zadania z tej listy. Uwaga: zadania z uławkami i SAM-em są bardzo łatwe.

Zadanie 1.(1pkt)(*) Rozwiąż zadanie z zeszłych list, jakiego jeszcze nie robiłeś (lub wykonaj operację upgrade na zadaniu za 0.5).

Zadanie 2.(1pkt) W zadaniu tym będziesz symulował Adama Mickiewicza piszącego Pana Tadeusza. Powinieneś wczytać tekst P. Tadeusza i następnie utworzyć strukturę, która dla każdego słowa pamięta listę możliwych następców tego słowa (czyli słów, które po tym występują). Fragment tej struktury może wyglądać następująco:

```
{ ..., 'natenczas' : ['wojski', 'z', 'i', 'tam'], 'kociołkach' : ['bigos'],  
  'cymbalistów' : ['wielu'], ...}
```

Uznajemy znaki interpunkcyjne za prawidłowe wyrazy, nie przejmujemy się ponadto przechodzeniem do nowego wiersza. Symulator Adama Mickiewicza (SAM) działa tak: przyjmuje na wejście jakieś słowo, znajduje jego następników, losuje jednego z nich, i czynności powtarza, aż do otrzymania tekstu o pożądanej długości (albo do otrzymania słowa bez następnika).

Za zadanie jest premia, 0.2, jeżeli będziesz uwzględniał nie tylko następstwa wyrazów (czyli pary wyrazów), ale również ich trójki, to znaczy, że to, jaki wyraz wylosujemy powinno zależeć od dwóch poprzednich (a tylko jeżeli nie da się znaleźć następstwa dla dwóch wyrazów, korzysta z poprzednio omówionej metody).

Zadanie 3.(1pkt) *Zadanie z przelewaniem* definiujemy w następujący sposób: mamy dwa wiadra (o pojemności X i Y litrów), początkowo puste. Celem jest doprowadzenie do sytuacji, w której w którymś wiadrze znajdzie się K litrów. Można wykonywać następujące ruchy:

- Napełnić jedno wiadro (czyli doprowadzić do sytuacji, w której znajduje się w nim dokładnie tyle wody, ile wynosi jego pojemność).
- Wylać całą wodę z wiadra na podłogę.
- Przeleć wodę z wiadra A do wiadra B . Można przeleć albo całą wodę (jeżeli pojemność i napełnienie wiadra B na to pozwala), albo też przeleć dokładnie tyle wody, żeby napełnić całkowicie wiadro B (wówczas reszta wody pozostaje w wiadrze A).

Wykorzystaj przeszukiwanie grafu w głąb do stworzenia funkcji, która dla zadanych pojemności wiader i zadanego celu zwraca ciąg stanów, prowadzący do sukcesu (czyli do stanu z co najmniej jednym wiadrzem zawierającym K litrów). Jeżeli taki ciąg nie istnieje, funkcja powinna zwracać listę pustą. Stany powinieneś reprezentować jako pary liczb (krotki lub listy dwuelementowe). Za zadanie można dostać 0.5 punktu premii, jeżeli rozwiąże się je w formie bardziej ogólnej (w której mamy N wiader).

Zadanie 4.(1pkt) Zdefiniuj klasę `Frac` (ułamki), która implementuje operacje na ułamkach. Klasa powinna mieć następujące właściwości.

- Dodawać, odejmować, dzielić i mnożyć w sposób dokładny ułamki.
- Używać do tych operacji naturalnej składni pythona (czyli operatorów `+-/*`).
- Wypisywać ułamki (w taki sposób, by działała również instrukcja `print [u1,u2,u3]`, gdzie `u1,u2,u3` są ułamkami).
- Porównywać ułamki.

Oczywiście nie możesz korzystać z klasy `Fraction` z biblioteki standardowej. Na KNO powinien pojawić się fragment definicji. Premia 0.5, jeżeli Twoje ułamki będą prawidłowo obsługiwać wartości $1/0$ ($+\infty$), $-1/0$ ($-\infty$) oraz $0/0$ (czyli Not-A-Number, `nan`).

Zadanie 5.(2pkt) Rozwiąż zadanie z grą, które pojawi się na KNO.