

# Wstęp do programowania

## Pracownia 9

### Publikacja: 7.12.2024

Premia za tę listę wynosi 0.5, przyznawana jest osobom, które zdobyły co najmniej 2p za zadania z tej listy. Maksimum dla tej listy wynosi 4p.

**Zadanie 1.(1pkt)** Napisz program, który wykorzystuje rekurencję do utworzenia ładnego rysunku. To może być kwadrat lub roślina, która znajdowała się w pierwszym zestawie rysunków na SKOSie, trójkąt lub dywan Sierpińskiego, drzewo Pitagorasa, lub też dowolny fraktal, który znajdziesz w Internecie. Oczywiście powinien to być rysunek, którego nie robiłeś.

**Zadanie 2.(1pkt)** Wróćmy do zadania z przestawianiem literek z imienia i nazwiska. Zmodyfikujemy je bardzo nieznacznie: z zadanych literek należy ułożyć nie dwa, lecz trzy słowa. Oznacza to w szczególności, że rozwiązanie, w którym znajdujemy słowa układalne i sprawdzamy wszystkie pary takich słów przestaje być akceptowalne (bo sprawdzenie wszystkich trójek trwa zbyt długo). Wskazówka: czy znając dwa słowa musisz przeglądać **wszystkie** układalne, żeby znaleźć trzecie? Wskazówka 2 (rot13): Cbzlfy b fybjavxh, j xgbelz xyhpmnzv fn cbfbegbjnar yvgrel quartb fybjn, n jnegbfpvn wrfg yvfgn fybj (wnxn?)

**Zadanie 3.(1pkt)** Wybierz wprawkę, której nie zrobiłeś. Może być z innej grupy, powinna mieć numer większy niż 2. Przez 'rozwiązanie wprawki' rozumiemy otrzymanie 1.7 punktu lub więcej (tzn. wprawka, której nie zrobiłeś to taka, w której zdobyłeś mniej niż 1.7 punktu)

**Zadanie 4.(1pkt)** Korzystając w dowolny sposób z programu `zycie.py` z wykładu 8 zaimplementuj inny automat komórkowy, mianowicie *Papier, nożyce, kamień*. Zasady są następujące:

1. Automat działa na prostokątnej planszy, wypełnionej kwadratowymi komórkami
2. Komórki sąsiadują ze sobą jeżeli stykają się bokami (4-sąsiedztwo, inaczej niż w 'życiu'). Nie ma zawijania planszy, komórki przy bokach po prostu mają mniej sąsiadów.
3. Mamy cztery rodzaje pól: pola puste oraz pola zawierające komórkę typu papier, nożyce lub kamień.
4. Pola niepuste dodatkowo mają zapisaną siłę (liczbę od 1 do 5).
5. Przy przejściu do nowego stanu kopiujemy planszę, a następnie dla każdego zajętego pola wykonujemy następującą operację: wybieramy losowo sąsiada i następnie:
  - a) Jeżeli sąsiadem jest pole puste, a nasza siła jest większa niż 1, to „zasiedlamy” je z siłą o 1 mniejszą (czyli na przykład papier z siłą 4 spowoduje wpisanie na puste miejsce papieru z siłą 3). Uwaga: podczas tej operacji patrzymy na starą planszę i modyfikujemy nową. Nie należy się przejmować, że pewne pola mogą zostać zmodyfikowane wiele razy
  - b) Jeżeli sąsiadem jest pole naszego koloru, to nic się nie dzieje.
  - c) Jeżeli sąsiadem jest pole innego rodzaju, to następuje pojedynek w wyniku którego przegrany traci jeden punkt siły, a zwycięzca zyskuje 1 (chyba, że już ma maksymalną siłę). Pojedynek rozstrzygany jest zgodnie z zasadami oryginalnej gry: papier pokonuje kamień, kamień nożyce, a te papier (siła nie ma znaczenia).
  - d) Jeżeli w wyniku pojedynku siła pola zmaleje do zera, pole staje się puste (jego zasiedlenie może nastąpić w kolejnym etapie, jeżeli tak się wylosuje).
6. Przypominamy, że odczyt pola wykonywany jest na „starej wersji planszy”, modyfikacje – na nowej, którą utworzyliśmy kopiując starą<sup>1</sup>.

**Napisz program, który implementuje te zasady.** Podobnie jak w programie `zycie.py` stan początkowy powinien być definiowany przez wielolinijkowy napis.

---

<sup>1</sup>Rozważ użycie funkcji `deepcopy` z modułu `copy`

**Rady do testowania:** Przygotowując plansze testowe uwzględnij następujące kwestie:

1. Jeżeli na planszy początkowej jest kilka pól (tylko) jednego rodzaju, wówczas powinny one zasiedlić swoją okolicę, po czym sytuacja się ustabilizuje
2. Jeżeli są dwa rodzaje pól, wówczas albo się nie spotkają, albo „ten silniejszy” zje w całości słabszego (na przykład „kamień” zje „nożyce”)
3. Rozważ wykonywanie testów za pomocą interpretera pypy (czyli zamiast pisać `python3 prog.py` pisz `pypy3 prog.py` lub `pypy prog.py`). Powinno to kilkukrotnie przyspieszyć działanie.
4. Niektóre plansze generują dramatyczne rozgrywki. Przykładowo poniższa:

```
...kkkkkkkkkkk.....
.....
.....
.....
..nnnn.....
..nnnn.....
..n.....
.....
.....kkkkkkkkkkk..
.....
.....
.....
.....ppppp.....
.....
```

Jeżeli nożycom nie uda się wyrwać z okrążenia, stosunkowo szybko wygrywa papier. Jeżeli się uda, to gra jest dość długa i może się skończyć zwycięstwem każdej strony (lub być może trwać w nieskończoność).