

Projektowanie algorytmów (w języku Python) – narzędzia programistyczne

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

Maciej Hojda

1 Zadania

1.1 Instalacja oprogramowania

1. Maszyna wirtualna Xubuntu (opcjonalne).
2. Python w wersji 3.8.5
3. Środowisko zintegrowane PyCharm.
4. Podstawowe biblioteki.

Uwaga: Korzystanie z wbudowanych i dodatkowych bibliotek Python-a będzie ograniczane (np. odpada skorzystanie z wbudowanej funkcji sortującej jeśli zadanie polega na implementacji algorytmu sortowania).

1.2 Zadanie nr 1

Uruchom (i przeanalizuj) następujący program

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph()
G.add_edge('A', 'B', weight=4)
G.add_edge('B', 'D', weight=2)
G.add_edge('A', 'C', weight=3)
G.add_edge('C', 'D', weight=4)

pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos, node_size = 500)
nx.draw_networkx_labels(G, pos)
nx.draw_networkx_edges(G, pos)
plt.show()
```

1.3 Zadanie nr 2

Uruchom (i przeanalizuj) następujący program

```
import networkx as nx
import matplotlib.pyplot as plt
```

```

import numpy as np

G = nx.Graph()
VV = [1, 2, 3, 4, 5]
WW = [(1, 2), (2, 3), (3, 4), (4, 5), (1, 3), (3, 5)]
Vx = {1:0, 2:1, 3:2, 4:3, 5:4}
Vy = {1:0, 2:1, 3:0, 4:-1, 5:0}

g = nx.Graph();
gpos = {};

for v in VV:
    g.add_node(v);
    gpos[v] = [Vx[v], Vy[v]]

for v1 in VV:
    for v2 in VV:
        if (v1, v2) in WW:
            label = str(np.sqrt((Vx[v1] - Vx[v2])**2 + (Vy[v1] - Vy[v2])**2))
            g.add_weighted_edges_from([(v1, v2, label)])

nx.draw(g, gpos, with_labels=True, node_color='yellow')
labels = nx.get_edge_attributes(g, 'weight')
np.draw_networkx_edge_labels(g, gpos, edge_labels=labels)

plt.show()

```

1.4 Zadanie nr 3

Napisz program wyświetlający graf pełny o parametrach: liczba wierzchołków zadana parametrycznie (jako stała w programie), wierzchołki rozmieszczone na okręgu, w równych odstępach, etykiety wierzchołków są kolejnymi liczbami naturalnymi.

1.5 Zadanie nr 4

Napisz program losujący graf o parametrycznie zadanej liczbie wierzchołków. Wierzchołki są rozmieszczone w losowych punktach (całego obszaru grafu) ale tak, żeby na siebie nie zachodziły. Wierzchołki dodawaj do grafu pojedynczo, jeśli wylosowany wierzchołek zachodzi na inny, to wylosuj nową pozycję dodawanego (powtarzaj tyle razy, aż nie będzie kolizji). Przerwij działanie programu, jeśli nie udało się wygenerować grafu po 100 nieudanych próbach dodania wierzchołka.

2 Maszyna wirtualna (opcjonalne)

Dla wygody korzystania z obowiązującego środowiska i wersji języka Python, można skorzystać z maszyny wirtualnej. Procedura instalacji jest jak następuje.

Uwaga: Wszystkie programy będą testowane na właśnie takiej maszynie.

2.1 Pobieranie

1. Pobierz i zainstaluj oprogramowanie VirtualBox (6.1)
<https://www.virtualbox.org/wiki/Downloads>
2. Pobierz obraz systemu operacyjnego Xubuntu (lts 20.04)
<https://xubuntu.org/download>

2.2 Tworzenie maszyny wirtualnej (Xubuntu)

1. Uruchom VirtualBox-a
2. Utwórz maszynę; menu: Machine -> New
Utwórz maszynę o parametrach
 - (a) Name: pwr
 - (b) Type: Linux
 - (c) Version: Ubuntu (64-bit)
 - (d) Memory size: 2048 MB
 - (e) Hard disk: create virtual hard disk now; VDI; dynamically allocated; 20GB
3. Ustaw opcje maszyny; prawy przycisk: Settings
 - (a) menu: Storage > Controller IDE > Empty
Z prawej strony: Attributes > Optical Drive > [ikona CD] > Choose a disk file
 - (b) Wybierz wcześniej pobrany obraz systemu.
 - (c) Zatwierdź
4. Uruchom maszynę
 - (a) Wybierz dysk startowy: zamontowany wcześniej obraz
 - (b) Install Xubuntu; wybierz język, klawiaturę; odznacz download updates; erase disk and install Xubuntu; wybierz lokalizację, podaj nazwę użytkownika i hasło
5. Zaktualizuj system
 - (a) Zrestartuj maszynę
 - (b) Uruchom konsolę: Ctrl+Alt+T
Uwaga: » oznacza komendę do wpisania w konsoli; « oznacza spodziewany rezultat wyświetlony w konsoli.
 - (c) >> `sudo apt update`
(podaj hasło)
 - (d) >> `sudo apt upgrade`
6. Ustaw reguły firewalla
 - (a) >> `sudo iptables -P INPUT DROP`
 - (b) >> `sudo iptables -P FORWARD DROP`
 - (c) >> `sudo ip6tables -P INPUT DROP`
 - (d) >> `sudo ip6tables -P FORWARD DROP`
 - (e) >> `sudo ip6tables -P OUTPUT DROP`
 - (f) >> `sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT`
 - (g) >> `sudo iptables -A INPUT -i lo -j ACCEPT`
 - (h) >> `sudo apt install iptables-persistent`
(zaakceptuj domyślne)
7. Zainstaluj Dodatki
 - (a) menu: Devices > Install Guest Addon CD image
 - (b) otwórz płytę z pulpitu; zapamiętaj ścieżkę
 - (c) uruchom konsolę, przejdź do ścieżki
 - (d) >> `cd /media/[...]`

- (e) >> sudo apt install build-essential module-assistant gcc make perl dkms
- (f) >> sudo ./autorun.sh
- (g) >> sudo usermod -aG vboxsf [nazwa użytkownika]
- (h) >> sudo reboot 0

3 Instalacja Pythona i PyCharma (przykład dla Xubuntu)

1. Zainstaluj Pythona

- (a) Ze strony
<https://www.python.org/downloads/release/python-385/>
pobierz Gzipped source tarball
- (b) >> sudo apt install build-essential checkinstall
libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev
tk-dev libgdbm-dev libc6-dev libbz2-dev libffi-dev zlib1g-dev
- (c) >> cd ~/Down*
- (d) >> tar -xvf Python*
- (e) >> cd Python*5
- (f) >> sudo ./configure --enable-optimizations
- (g) >> sudo make altinstall
zweryfikuj instalację
- (h) >> python3

2. Zainstaluj PyCharma

- (a) Ze strony
<https://www.jetbrains.com/pycharm/download/#section=linux>
pobierz PyCharm community edition
- (b) >> sudo apt install python3.8-distutils
- (c) >> cd ~/Down*
- (d) >> tar -xvf pycharm*
- (e) >> cd [rozpakowany katalog]\bin
- (f) >> chmod 777 pycharm.sh
- (g) >> ./pycharm.sh
- (h) >> sudo apt install python3-tk
- (i) >> sudo reboot 0

3. Zainstaluj Pakiety pomocnicze

- (a) Uruchom PyCharma
- (b) Utwórz nowy projekt
- (c) w konsoli pycharma (na dole: Python Console) wpisz:
- (d) >> import pip
- (e) >> pip.main(['install', 'numpy'])
- (f) >> pip.main(['install', 'networkx'])
- (g) >> pip.main(['install', 'matplotlib'])