

Michał Szczurek lab 6

```
In [1]: from collections import deque
import copy as np
import sys
import matplotlib.pyplot as plt
import time
from timeit import timeit
import random, string

1. Zaimplementuj algorytm wyszukiwania wzorca 2-wymiarowego

In [2]: class Node:
    def __init__(self, char, parent, node_id, accepting):
        self.children = {}
        self.accepting = accepting
        self.char = char
        self.id = node_id
        self.parent = parent
        self.fail_link = None
    def add_child(self, char):
        d[char] = Node(char)
    def has_child(self, char):
        return char in self.children
    def __str__(self):
        return str(self.char) + " " + str(self.id) + " " + str(self.accepting)

In [3]: class Automaton:
    def __init__(self, patterns):
        self.root = Node("S", None, 0, None)
        self.next_id = 1
        self.pattern_dict = {} # to store patterns ids
        self.patterns = patterns
        for i, p in enumerate(patterns):
            self.add(i, p)
        self.add_fail_links()
        self.transform()
    def add(self, index, text):
        i = 0
        node = self.root
        while i < len(text) and node.has_child(text[i]):
            node = node.children[text[i]]
            i += 1
        while i < len(text):
            node.children[text[i]] = Node(text[i], node, self.next_id, index if i == len(text) - 1 else None)
            if i == len(text) - 1:
                self.children[text[i]] = index
            node = node.children[text[i]]
            i += 1
        self.next_id += 1
    def add_fail_links(self):
        que = deque()
        for (key, node) in self.root.children.items():
            que.append(node)
        # first level fail links
        while len(que) > 0:
            cur = que.popleft()
            if cur.parent is not self.root:
                for node in cur.children.values():
                    que.append(node)
            cur.fail_link = self.root
        else: # while's else - Trie has only 1 level
            cur = None
        # other fail links
        while cur:
            # add children for later
            for node in cur.children.values():
                que.append(node)
            # build fail link
            char = cur.char
            prev = cur.parent.fail_link
            while char not in prev.children and prev is not self.root:
                prev = prev.fail_link
            # "joker" from root case
            if prev is self.root and char not in self.root.children:
                curr.fail_link = self.root
            else:
                curr.fail_link = prev.children[char]
            if len(que) > 0:
                cur = que.popleft()
            else:
                cur = None
    def transform(self):
        """Takes tree with fail links and returns tree with "shortcut" edges"""
        que = deque()
        node = self.root.children.values()
        que.append(node)
        while len(que) > 0:
            cur = que.popleft()
            for node in cur.children.values():
                que.append(node)
            for char, child in cur.fail_link.children.items():
                if char not in cur.children:
                    cur.children[char] = child
    def search(self, text):
        res = [-1 for _ in text]
        state = self.root
        for i, char in enumerate(text):
            if char not in state.children:
                state = self.root
            else:
                state = state.children[char]
            if state.accepting == None:
                res[i] = state.accepting
        return res

In [4]: def search_2d(text, patterns):
    """Returns coordinates of lower right symbol of pattern"""
    pattern_set = set(patterns)
    automaton_id = Automaton(pattern_set)
    res_id = [automaton_id.search(line) for line in text]
    res_id = np.array(res_id).transpose()
    pattern = tuple([automaton_id.pattern_dict[p] for p in patterns]) # pattern made of patterns using ids
    automaton_2d = Automaton(pattern)
    res_2d = [automaton_2d.search(line) for line in res_id]
    res = []
    for i in range(len(res_2d)):
        for j in range(len(res_2d[i])):
            if res_2d[i][j] != -1:
                res.append((j, i))
    return res
```

2. Znajdź w załączonym pliku "haystack.txt" wszystkie sytuacje, gdy taka sama litera występuje na tej samej pozycji w dwóch kolejnych liniach. Zwróć uwagę, na nierówną długość linii w pliku.

Funkcja dodająca do tekstu meta-znak, tak by tekst był prostokątny

```
In [5]: def parse_text(text, null_symbol='S'):
    max_len = 0
    for i in text:
        max_len = max(max_len, len(i))
    res = []
    for line in text:
        res.append(line + str((max_len - len(line)) * null_symbol))
    return res

In [6]: with open("haystack.txt", "r", encoding='utf-8') as file:
    text = file.readlines()
```

Wyszukiwanie sytuacji z polecenia

```
In [7]: letters = set()
for line in text:
    for c in line:
        letters.add(c)

total = 0
letters = sorted(letters)
for letter in letters:
    res = search_2d(parse_text(text), [letter, letter])
    print(f"({repr(letter)}) Liczba sytuacji: {len(res)} : {res}")
    total += len(res)
print(f"Suma {total} sytuacji")

\n" Liczba sytuacji: 1 : [(38, 79)]
\n" Liczba sytuacji: 158 : [(28, 1), (38, 1), (83, 1), (83, 1), (2, 2), (5, 2), (59, 2), (81, 2), (20, 3), (21, 3), (28, 3), (29, 3), (58, 3), (28, 8), (38, 8), (9, 7), (6, 7), (11, 7), (38, 7), (39, 7), (58, 7), (25, 8), (34, 8), (41, 8), (5, 8), (6, 8), (1, 18), (1, 18), (2, 18), (53, 18), (11, 11), (16, 11), (21, 11), (47, 12), (58, 12), (23, 13), (45, 13), (64, 13), (64, 13), (22, 15), (23, 15), (42, 15), (42, 15), (62, 15), (25, 16), (46, 16), (78, 16), (13, 17), (61, 18), (3, 19), (23, 19), (24, 19), (25, 19), (46, 19), (59, 19), (52, 21), (57, 21), (23, 22), (81, 22), (31, 23), (48, 23), (49, 23), (14, 24), (1, 24), (9, 25), (58, 26), (48, 26), (71, 26), (88, 26), (1, 27), (82, 27), (81, 28), (54, 29), (57, 29), (63, 29), (69, 29), (29, 30), (64, 31), (19, 32), (46, 32), (47, 32), (55, 32), (72, 32), (36, 3), (42, 33), (43, 33), (58, 33), (2, 34), (59, 34), (44, 35), (49, 35), (49, 35), (55, 36), (24, 38), (21, 39), (72, 39), (40, 4), (41, 23), (42, 23), (43, 23), (44, 23), (45, 23), (46, 23), (47, 23), (48, 23), (49, 23), (50, 23), (51, 23), (52, 23), (53, 23), (54, 23), (55, 23), (56, 23), (57, 23), (58, 23), (59, 23), (60, 23), (61, 23), (62, 23), (63, 23), (64, 23), (65, 23), (66, 23), (67, 23), (68, 23), (69, 23), (70, 23), (71, 23), (72, 23), (73, 23), (74, 23), (75, 23), (76, 23), (77, 23), (78, 23), (79, 23), (80, 23), (81, 23), (82, 23), (83, 23), (84, 23), (85, 23), (86, 23), (87, 23), (88, 23), (89, 23), (90, 23), (91, 23), (92, 23), (93, 23), (94, 23), (95, 23), (96, 23), (97, 23), (98, 23), (99, 23), (100, 23), (101, 23), (102, 23), (103, 23), (104, 23), (105, 23), (106, 23), (107, 23), (108, 23), (109, 23), (110, 23), (111, 23), (112, 23), (113, 23), (114, 23), (115, 23), (116, 23), (117, 23), (118, 23), (119, 23), (120, 23), (121, 23), (122, 23), (123, 23), (124, 23), (125, 23), (126, 23), (127, 23), (128, 23), (129, 23), (130, 23), (131, 23), (132, 23), (133, 23), (134, 23), (135, 23), (136, 23), (137, 23), (138, 23), (139, 23), (140, 23), (141, 23), (142, 23), (143, 23), (144, 23), (145, 23), (146, 23), (147, 23), (148, 23), (149, 23), (150, 23), (151, 23), (152, 23), (153, 23), (154, 23), (155, 23), (156, 23), (157, 23), (158, 23), (159, 23), (160, 23), (161, 23), (162, 23), (163, 23), (164, 23), (165, 23), (166, 23), (167, 23), (168, 23), (169, 23), (170, 23), (171, 23), (172, 23), (173, 23), (174, 23), (175, 23), (176, 23), (177, 23), (178, 23), (179, 23), (180, 23), (181, 23), (182, 23), (183, 23), (184, 23), (185, 23), (186, 23), (187, 23), (188, 23), (189, 23), (190, 23), (191, 23), (192, 23), (193, 23), (194, 23), (195, 23), (196, 23), (197, 23), (198, 23), (199, 23), (200, 23), (201, 23), (202, 23), (203, 23), (204, 23), (205, 23), (206, 23), (207, 23), (208, 23), (209, 23), (210, 23), (211, 23), (212, 23), (213, 23), (214, 23), (215, 23), (216, 23), (217, 23), (218, 23), (219, 23), (220, 23), (221, 23), (222, 23), (223, 23), (224, 23), (225, 23), (226, 23), (227, 23), (228, 23), (229, 23), (230, 23), (231, 23), (232, 23), (233, 23), (234, 23), (235, 23), (236, 23), (237, 23), (238, 23), (239, 23), (240, 23), (241, 23), (242, 23), (243, 23), (244, 23), (245, 23), (246, 23), (247, 23), (248, 23), (249, 23), (250, 23), (251, 23), (252, 23), (253, 23), (254, 23), (255, 23), (256, 23), (257, 23), (258, 23), (259, 23), (260, 23), (261, 23), (262, 23), (263, 23), (264, 23), (265, 23), (266, 23), (267, 23), (268, 23), (269, 23), (270, 23), (271, 23), (272, 23), (273, 23), (274, 23), (275, 23), (276, 23), (277, 23), (278, 23), (279, 23), (280, 23), (281, 23), (282, 23), (283, 23), (284, 23), (285, 23), (286, 23), (287, 23), (288, 23), (289, 23), (290, 23), (291, 23), (292, 23), (293, 23), (294, 23), (295, 23), (296, 23), (297, 23), (298, 23), (299, 23), (300, 23), (301, 23), (302, 23), (303, 23), (304, 23), (305, 23), (306, 23), (307, 23), (308, 23), (309, 23), (310, 23), (311, 23), (312, 23), (313, 23), (314, 23), (315, 23), (316, 23), (317, 23), (318, 23), (319, 23), (320, 23), (321, 23), (322, 23), (323, 23), (324, 23), (325, 23), (326, 23), (327, 23), (328, 23), (329, 23), (330, 23), (331, 23), (332, 23), (333, 23), (334, 23), (335, 23), (336, 23), (337, 23), (338, 23), (339, 23), (340, 23), (341, 23), (342, 23), (343, 23), (344, 23), (345, 23), (346, 23), (347, 23), (348, 23), (349, 23), (350, 23), (351, 23), (352, 23), (353, 23), (354, 23), (355, 23), (356, 23), (357, 23), (358, 23), (359, 23), (360, 23), (361, 23), (362, 23), (363, 23), (364, 23), (365, 23), (366, 23), (367, 23), (368, 23), (369, 23), (370, 23), (371, 23), (372, 23), (373, 23), (374, 23), (375, 23), (376, 23), (377, 23), (378, 23), (379, 23), (380, 23), (381, 23), (382, 23), (383, 23), (384, 23), (385, 23), (386, 23), (387, 23), (388, 23), (389, 23), (390, 23), (391, 23), (392, 23), (393, 23), (394, 23), (395, 23), (396, 23), (397, 23), (398, 23), (399, 23), (400, 23), (401, 23), (402, 23), (403, 23), (404, 23), (405, 23), (406, 23), (407, 23), (408, 23), (409, 23), (410, 23), (411, 23), (412, 23), (413, 23), (414, 23), (415, 23), (416, 23), (417, 23), (418, 23), (419, 23), (420, 23), (421, 23), (422, 23), (423, 23), (424, 23), (425, 23), (426, 23), (427, 23), (428, 23), (429, 23), (430, 23), (431, 23), (432, 23), (433, 23), (434, 23), (435, 23), (436, 23), (437, 23), (438, 23), (439, 23), (440, 23), (441, 23), (442, 23), (443, 23), (444, 23), (445, 23), (446, 23), (447, 23), (448, 23), (449, 23), (450, 23), (451, 23), (452, 23), (453, 23), (454, 23), (455, 23), (456, 23), (457, 23), (458, 23), (459, 23), (460, 23), (461, 23), (462, 23), (463, 23), (464, 23), (465, 23), (466, 23), (467, 23), (468, 23), (469, 23), (470, 23), (471, 23), (472, 23), (473, 23), (474, 23), (475, 23), (476, 23), (477, 23), (478, 23), (479, 23), (480, 23), (481, 23), (482, 23), (483, 23), (484, 23), (485, 23), (486, 23), (487, 23), (488, 23), (489, 23), (490, 23), (491, 23), (492, 23), (493, 23), (494, 23), (495, 23), (496, 23), (497, 23), (498, 23), (499, 23), (500, 23), (501, 23), (502, 23), (503, 23), (504, 23), (505, 23), (506, 23), (507, 23), (508, 23), (509, 23), (510, 23), (511, 23), (512, 23), (513, 23), (514, 23), (515, 23), (516, 23), (517, 23), (518, 23), (519, 23), (520, 23), (521, 23), (522, 23), (523, 23), (524, 23), (525, 23), (526, 23), (527, 23), (528, 23), (529, 23), (530, 23), (531, 23), (532, 23), (533, 23), (534, 23), (535, 23), (536, 23), (537, 23), (538, 23), (539, 23), (540, 23), (541, 23), (542, 23), (543, 23), (544, 23), (545, 23), (546, 23), (547, 23), (548, 23), (549, 23), (550, 23), (551, 23), (552, 23), (553, 23), (554, 23), (555, 23), (556, 23), (557, 23), (558, 23), (559, 23), (560, 23), (561, 23), (562, 23), (563, 23), (564, 23), (565, 23), (566, 23), (567, 23), (568, 23), (569, 23), (570, 23), (571, 23), (572, 23), (573, 23), (574, 23), (575, 23), (576, 23), (577, 23), (578, 23), (579, 23), (580, 23), (581, 23), (582, 23), (583, 23), (584, 23), (585, 23), (586, 23), (587, 23), (588, 23), (589, 23), (590, 23), (591, 23), (592, 23), (593, 23), (594, 23), (595, 23), (596, 23), (597, 23), (598, 23), (599, 23), (600, 23), (601, 23), (602, 23), (603, 23), (604, 23), (605, 23), (606, 23), (607, 23), (608, 23), (609, 23), (610, 23), (611, 23), (612, 23), (613, 23), (614, 23), (615, 23), (616, 23), (617, 23), (618, 23), (619, 23), (620, 23), (621, 23), (622, 23), (623, 23), (624, 23), (625, 23), (626, 23), (627, 23), (628, 23), (629, 23), (630, 23), (631, 23), (632, 23), (633, 23), (634, 23), (635, 23), (636, 23), (637, 23), (638, 23), (639, 23), (640, 23), (641, 23), (642, 23), (643, 23), (644, 23), (645, 23), (646, 23), (647, 23), (648, 23), (649, 23), (650, 23), (651, 23), (652, 23), (653, 23), (654, 23), (655, 23), (656, 23), (657, 23), (658, 23), (659, 23), (660, 23), (661, 23), (662, 23), (663, 23), (664, 23), (665, 23), (666, 23), (667, 23), (668, 23), (669, 23), (670, 23), (671, 23), (672, 23), (673, 23), (674, 23), (675, 23), (676, 23), (677, 23), (678, 23), (679, 23), (680, 23), (681, 23), (682, 23), (683, 23), (684, 23), (685, 23), (686, 23), (687, 23), (688, 23), (689, 23), (690, 23), (691, 23), (692, 23), (693, 23), (694, 23), (695, 23), (696, 23), (697, 23), (698, 23), (699, 23), (700, 23), (701, 23), (702, 23), (703, 23), (704, 23), (705, 23), (706, 23), (707, 23), (708, 23), (709, 23), (710, 23), (711, 23), (712, 23), (713, 23), (714, 23), (715, 23), (716, 23), (717, 23), (718, 23), (719, 23), (720, 23), (721, 23), (722, 23), (723, 23), (724, 23), (725, 23), (726, 23), (727, 23), (728, 23), (729, 23), (730, 23), (731, 23), (732, 23), (733, 23), (734, 23), (735, 23), (736, 23), (737, 23), (738, 23), (739, 23), (740, 23), (741, 23), (742, 23), (743, 23), (744, 23), (745, 23), (746, 23), (747, 23), (748, 23), (749, 23), (750, 23), (751, 23), (752, 23), (753, 23), (754, 23), (755, 23), (756, 23), (757, 23), (758, 23), (759, 23), (760, 23), (761, 23), (762, 23), (763, 23), (764, 23), (765, 23), (766, 23), (767, 23), (768, 23), (769, 23), (770, 23), (771, 23), (772, 23), (773, 23), (774, 23), (775, 23), (776, 23), (777, 23), (778, 23), (779, 23), (780, 23), (781, 23), (782, 23), (783, 23), (784, 23), (785, 23), (786, 23), (787, 23), (788, 23), (789, 23), (790, 23), (791, 23), (792, 23), (793, 23), (794, 23), (795, 23), (796, 23), (797, 23), (798, 23), (799, 23), (800, 23), (801, 23), (802, 23), (803, 23), (804, 23), (805, 23), (806, 23), (807, 23), (808, 23), (809, 23), (810, 23), (811, 23), (812, 23), (813, 23), (814, 23), (815, 23), (816, 23), (817, 23), (818, 23), (819, 23), (820, 23), (821, 23), (822, 23), (823, 23), (824, 23), (825, 23), (826, 23), (827, 23), (828, 23), (829, 23), (830, 23), (831, 23), (832, 23), (833, 23), (834, 23), (835, 23), (836, 23), (837, 23), (838, 23), (839, 23), (840, 23), (841, 23), (842, 23), (843, 23), (844, 23), (845, 23), (846, 23), (847, 23), (848, 23), (849, 23), (850, 23), (851, 23), (852, 23), (853, 23), (854, 23), (855, 23), (856, 23), (857, 23), (858, 23), (859, 23), (860, 23), (861, 23), (862, 23), (863, 23), (864, 23), (865, 23), (866, 23), (867, 23), (868, 23), (869, 23), (870, 23), (871, 23), (872, 23), (873, 23), (874, 23), (875, 23), (876, 23), (877, 23), (878, 23), (879, 23), (880, 23), (881, 23), (882, 23), (883, 23), (884, 23), (885, 23), (886, 23), (887, 23), (888, 23), (889, 23), (890, 23), (891, 23), (892, 23), (893, 23), (894, 23), (895, 23), (896, 23), (897, 23), (898, 23), (899, 23), (900, 23), (901, 23), (902, 23), (903, 23), (904, 23), (905, 23), (906, 23), (907, 23), (908, 23), (909, 23), (910, 23), (911, 23), (912, 23), (913, 23), (914, 23), (915, 23), (916, 23), (917, 23), (918, 23), (919, 23), (920, 23), (921, 23), (922, 23), (923, 23), (924, 23), (925, 23), (926, 23), (927, 23), (928, 23), (929, 23), (930, 23), (931, 23), (932, 23), (933, 23), (934, 23), (935, 23), (936, 23), (937, 23), (938, 23), (939, 23), (940, 23), (941, 23), (942, 23), (943, 23), (944, 23), (945, 23), (946, 23), (947, 23), (948, 23), (949, 23), (950, 23), (951, 23), (952, 23), (953, 23), (954, 23), (955, 23), (956, 23), (957, 23), (958, 23), (959, 23), (960, 23), (961, 23), (962, 23), (963, 23), (964, 23), (965, 23), (966, 23), (967, 23), (968, 23), (969, 23), (970, 23), (971, 23), (972, 23), (973, 23), (974, 23), (975, 23), (976, 23), (977, 23), (978, 23), (979, 23), (980, 23), (981, 23), (982, 23), (983, 23), (984, 23), (985, 23), (986, 23), (987, 23), (988, 23), (989, 23), (990, 23), (991, 23), (992, 23), (993, 23), (994, 23), (995, 23), (996, 23), (997, 23), (998, 23), (999, 23), (1000, 23), (1001, 23), (1002, 23), (1003, 23), (1004, 23), (1005, 23), (1006, 23), (1007, 23), (1008, 23), (1009, 23), (1010, 23), (1011, 23), (1012, 23), (1013, 23), (1014, 23), (1015, 23), (1016, 23), (1017, 23), (1018, 23), (1019, 23), (1020, 23), (1021, 23), (1022, 23), (1023, 23), (1024, 23), (1025, 23), (1026, 23), (1027, 23), (1028, 23), (1029, 23), (1030, 23), (1031, 23), (1032, 23), (1033, 23), (1034, 23), (1035, 23), (1036, 23), (1037, 23), (1038, 23), (1039, 23), (1040, 23), (1041, 23), (1042, 23), (1043, 23), (1044, 23), (1045, 23), (1046, 23), (1047, 23), (1048, 23), (1049, 23), (1050, 23), (1051, 23), (1052, 23), (1053, 23), (1054, 23), (1055, 23), (1056, 23), (1057, 23), (1058, 23), (1059, 23), (1060, 23), (1061, 23), (1062, 23), (1063, 23), (1064, 23), (1065, 23), (1066, 23), (1067, 23), (1068, 23), (1069, 23), (1070, 23), (1071, 23), (1072, 23), (1073, 23), (1074, 23), (1075, 23), (1076, 23), (1077, 23), (1078, 23), (1079, 23), (1080, 23), (1081, 23), (1082, 23), (1083, 23), (1084, 23), (1085, 23), (1086, 23), (1087, 23), (1088, 23), (1089, 23), (1090, 23), (1091, 23), (1092, 23), (1093, 23), (1094, 23), (1095, 23), (1096, 23), (1097, 23), (1098, 23), (1099, 23), (1100, 23), (1101, 23), (1102, 23), (1103, 23), (1104, 23), (1105, 23), (1106, 23), (1107, 23), (1108, 23), (1109, 23), (1110, 23), (1111, 23), (1112, 23), (1113, 23), (1114, 23), (1115, 23), (1116, 23), (1117, 23), (1118, 23), (1119, 23), (1120, 23), (1121, 23), (1122, 23), (1123, 23), (1124, 23), (1125, 23), (1126, 23), (1127, 23), (1128, 23), (1129, 23), (1130, 23), (1131, 23), (1132, 23), (1133, 23), (1134, 23), (1135, 23), (1136, 23), (1137, 23), (1138, 23), (1139, 23), (1140, 23), (1141, 23), (1142, 23), (1143, 23), (1144, 23), (
```