Jakub Janicki
28.05.2021

Laboratorium 6 – Wyszukiwanie wzorca 2d

1. *Została zaimplementowana struktura automatu na podstawie algorytmu Aho-Corasick. Został zaimplementowany algorytm wyszukiwania wzorca 2d na z użyciem automatu*

```
In [60]: text = ["ccdas",
                 "ccccd",
                 "cdccc",
                 "dacdc"]

         pattern = ["ccd", "ccc", "cdc"]

         search_2d(text, pattern)

Out[60]: [(0, 0), (1, 2)]
```

Rysunek 1 Test wyszukiwania 2d

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

Test przeprowadzono dla wszystkich liter alfabetu angielskiego. Poniższe wyniki opisują liczbę wystąpień tego warunku oraz określają miejsca wystąpienia. Żadna z dużych liter nie spełnia warunku

```
Letter: a-a appeared 28 times
 [(1, 82), (4, 30), (6, 60), (7, 63), (21, 6), (29, 69), (32, 50), (32, 73), (34, 66), (38, 4), (53, 12), (54, 12), (54, 4
8), (57, 11), (58, 36), (59, 36), (60, 24), (65, 2), (65, 14), (65, 22), (66, 35), (70, 35), (77, 21), (77, 74), (78, 42),
(78, 61), (79, 59), (80, 37)]
Letter: b-b appeared 0 times
 []
Letter: c-c appeared 6 times
 [(4, 54), (11, 45), (14, 10), (42, 0), (69, 0), (83, 41)]
Letter: d-d appeared 1 times
 [(38, 19)]
Letter: e-e appeared 48 times
 [(1, 63), (2, 8), (5, 77), (8, 65), (11, 1), (11, 64), (15, 2), (16, 43), (18, 6), (19, 27), (21, 10), (22, 61), (23, 53),
(25, 3), (25, 65), (29, 67), (29, 73), (30, 38), (30, 43), (38, 48), (41, 11), (41, 26), (42, 57), (43, 36), (43, 48), (47,
52), (48, 50), (52, 31), (58, 54), (59, 50), (59, 54), (60, 73), (64, 66), (66, 69), (67, 72), (68, 17), (69, 46), (70, 1
5), (71, 49), (72, 38), (73, 23), (74, 27), (77, 6), (78, 6), (79, 65), (81, 6), (82, 14), (83, 47)]
Letter: f-f appeared 2 times
 [(31, 59), (78, 1)]
Letter: g-g appeared 0 times
 []
Letter: h-h appeared 4 times
 [(28, 2), (38, 2), (57, 31), (74, 12)]
Letter: i-i appeared 13 times
 [(2, 5), (9, 37), (10, 60), (20, 55), (32, 0), (32, 31), (45, 33), (53, 69), (56, 17), (61, 45), (69, 51), (74, 13), (78,
13)]
Letter: j-j appeared 0 times
 []
Letter: k-k appeared 0 times
 []
Letter: l-l appeared 5 times
 [(29, 72), (34, 45), (42, 77), (47, 61), (54, 45)]
Letter: m-m appeared 5 times
 [(17, 5), (29, 70), (35, 40), (35, 60), (45, 0)]
Letter: n-n appeared 15 times
 [(1, 83), (2, 9), (15, 54), (20, 37), (21, 56), (22, 62), (32, 1), (36, 18), (52, 32), (55, 33), (57, 13), (65, 29), (68,
35), (68, 40), (68, 57)]
Letter: o-o appeared 21 times
 [(5, 75), (6, 66), (7, 38), (8, 38), (11, 27), (16, 60), (28, 17), (29, 17), (31, 58), (33, 34), (34, 11), (34, 26), (42,
1), (45, 55), (51, 2), (53, 8), (54, 1), (59, 45), (72, 42), (80, 10), (82, 52)]
Letter: p-p appeared 2 times
 [(29, 71), (42, 18)]
Letter: q-q appeared 0 times
 []
Letter: r-r appeared 21 times
 [(2, 4), (7, 39), (7, 50), (8, 13), (16, 18), (18, 14), (20, 54), (21, 54), (29, 65), (32, 70), (34, 10), (34, 37), (44, 2
5), (47, 42), (48, 37), (53, 5), (56, 40), (61, 30), (63, 39), (68, 29), (70, 22)]
Letter: s-s appeared 19 times
 [(4, 57), (4, 63), (5, 49), (9, 21), (10, 58), (29, 45), (30, 56), (31, 56), (38, 34), (41, 63), (46, 34), (47, 44), (50,
14), (53, 53), (55, 0), (68, 37), (71, 41), (72, 24), (80, 24)]
Letter: t-t appeared 41 times
 [(1, 7), (2, 6), (2, 7), (4, 37), (5, 14), (5, 23), (8, 29), (9, 75), (14, 55), (16, 12), (17, 3), (20, 33), (23, 8), (24,
4), (25, 4), (28, 31), (29, 23), (29, 52), (31, 16), (36, 10), (38, 0), (42, 45), (42, 73), (47, 24), (51, 0), (52, 33), (5
3, 61), (55, 11), (56, 54), (59, 49), (59, 78), (60, 33), (60, 75), (62, 56), (68, 71), (70, 5), (72, 3), (73, 3), (73, 1
0), (73, 59), (78, 22)]
Letter: u-u appeared 0 times
 []
Letter: v-v appeared 0 times
 []
Letter: w-w appeared 2 times
 [(2, 3), (22, 70)]
Letter: x-x appeared 1 times
 [(29, 68)]
Letter: y-y appeared 1 times
 [(45, 5)]
```

Rysunek 2 Zad2 wyniki

3. *Znajdź wszystkie wystąpienia "th" oraz "t h" w dwóch kolejnych liniach na tej samej pozycji.*

```
print(f"[ th ] appeared {len(search_2d(lines, ['th', 'th']))} times \n {search_2d(lines, ['th', 'th'])}")
print(f"[ t h ] appeared {len(search_2d(lines, ['t h', 't h']))} times \n {search_2d(lines, ['t h', 't h'])}")
```

```
[ th ] appeared 0 times
 []
[ t h ] appeared 1 times
 [(36, 0)]
```

Rysunek 3 Zad3 wyniki

4. *Wybierz przynajmniej 3 litery (małe). Znajdź wszystkie wystąpienia tej litery w załączonym pliku "haystack.png"*

Zostały obliczone wystąpienia liter a,s,m,o, a także zaprezentowane współrzędne wystąpienia litery a.

```
Letter: a appeared 356 times
Letter: s appeared 334 times
Letter: m appeared 131 times
Letter: o appeared 310 times
```

Rysunek 4 Zad4 Wyniki

```
Letter: a appeared 356 times
 [(37, 206), (37, 262), (37, 322), (37, 486), (37, 622), (37, 750), (59, 302), (59, 332), (59, 390), (59, 641), (59, 699),
(81, 176), (81, 202), (81, 398), (81, 564), (81, 588), (103, 55), (103, 104), (103, 271), (125, 273), (125, 327), (125, 61
8), (147, 125), (147, 155), (147, 246), (147, 353), (147, 548), (147, 631), (169, 55), (169, 89), (169, 193), (169, 321),
(169, 517), (169, 556), (169, 585), (169, 707), (191, 83), (191, 257), (191, 405), (191, 576), (191, 620), (191, 679), (21
3, 434), (213, 535), (235, 119), (235, 471), (235, 592), (257, 93), (257, 445), (279, 62), (279, 125), (279, 268), (279, 45
9), (279, 562), (301, 36), (323, 87), (323, 175), (323, 434), (323, 494), (323, 572), (345, 97), (345, 221), (345, 272), (3
45, 408), (345, 576), (367, 25), (367, 247), (367, 273), (367, 603), (367, 656), (389, 119), (389, 227), (389, 369), (389,
704), (411, 25), (433, 247), (433, 282), (433, 511), (433, 593), (433, 654), (455, 83), (455, 127), (455, 244), (455, 366),
(455, 448), (477, 25), (477, 81), (477, 166), (477, 200), (477, 308), (477, 564), (477, 712), (499, 77), (499, 159), (499,
444), (499, 694), (499, 743), (521, 64), (521, 147), (521, 196), (521, 224), (521, 451), (543, 54), (543, 126), (543, 258),
(543, 358), (543, 528), (543, 696), (565, 25), (565, 163), (565, 275), (565, 446), (565, 495), (565, 629), (587, 158), (58
7, 258), (587, 415), (587, 512), (587, 552), (587, 749), (631, 96), (631, 264), (631, 407), (631, 568), (653, 55), (653, 21
7), (653, 296), (653, 418), (653, 547), (653, 654), (675, 25), (675, 89), (675, 380), (675, 641), (697, 92), (697, 469), (6
97, 640), (697, 673), (719, 260), (719, 462), (719, 643), (741, 194), (741, 462), (741, 536), (741, 667), (763, 332), (763,
579), (763, 672), (763, 732), (785, 30), (785, 74), (785, 459), (785, 503), (785, 551), (785, 613), (785, 646), (807, 110),
(807, 216), (807, 629), (829, 280), (829, 322), (829, 406), (829, 600), (851, 55), (851, 242), (873, 55), (873, 368), (873,
640), (895, 105), (895, 225), (917, 228), (939, 82), (939, 473), (939, 503), (961, 168), (961, 219), (961, 389), (961, 69
4), (983, 568), (983, 610), (1005, 43), (1005, 420), (1005, 667), (1027, 166), (1027, 355), (1027, 433), (1049, 36), (1049,
170), (1049, 200), (1049, 439), (1049, 535), (1049, 584), (1071, 294), (1071, 324), (1071, 554), (1071, 671), (1093, 99),
(1093, 469), (1115, 435), (1115, 637), (1115, 727), (1115, 803), (1137, 79), (1137, 109), (1137, 192), (1159, 164), (1181,
49), (1181, 137), (1181, 159), (1181, 181), (1181, 225), (1181, 350), (1181, 643), (1203, 131), (1203, 291), (1203, 397),
(1203, 450), (1203, 540), (1225, 147), (1225, 350), (1225, 410), (1225, 443), (1225, 487), (1225, 588), (1225, 631), (1225,
661), (1225, 737), (1247, 261), (1247, 302), (1247, 374), (1269, 77), (1269, 135), (1269, 212), (1269, 565), (1291, 119),
(1291, 176), (1291, 354), (1291, 413), (1291, 441), (1291, 603), (1291, 663), (1313, 44), (1313, 202), (1313, 258), (1313,
328), (1313, 350), (1313, 371), (1313, 439), (1313, 698), (1335, 48), (1335, 169), (1335, 231), (1335, 330), (1335, 378),
(1335, 577), (1357, 63), (1357, 149), (1357, 230), (1357, 276), (1357, 574), (1357, 640), (1379, 25), (1379, 121), (1379, 3
48), (1379, 567), (1379, 637), (1401, 131), (1401, 175), (1423, 25), (1423, 343), (1445, 46), (1445, 145), (1445, 229), (14
45, 522), (1445, 567), (1467, 52), (1467, 152), (1467, 219), (1467, 268), (1467, 334), (1467, 487), (1467, 571), (1489, 18
1), (1489, 337), (1489, 487), (1511, 99), (1511, 223), (1511, 455), (1511, 499), (1511, 544), (1511, 671), (1533, 60), (153
3, 362), (1555, 35), (1555, 92), (1555, 140), (1555, 290), (1555, 345), (1577, 344), (1577, 488), (1599, 178), (1599, 321),
(1599, 527), (1621, 154), (1621, 337), (1621, 677), (1643, 36), (1643, 114), (1643, 291), (1643, 483), (1643, 501), (1643,
634), (1665, 79), (1665, 661), (1687, 59), (1709, 144), (1709, 166), (1709, 217), (1709, 399), (1709, 687), (1731, 190), (1
731, 225), (1731, 283), (1731, 301), (1731, 342), (1731, 386), (1731, 437), (1731, 507), (1731, 558), (1731, 616), (1731, 6
84), (1753, 308), (1753, 389), (1753, 496), (1753, 554), (1753, 572), (1775, 25), (1775, 76), (1775, 405), (1775, 508), (17
75, 560), (1797, 68), (1797, 188), (1797, 307), (1797, 365), (1797, 383), (1797, 625), (1797, 681), (1819, 238), (1819, 40
5), (1841, 217), (1841, 407), (1841, 497), (1863, 25), (1863, 71), (1863, 385)]
```

Rysunek 5 Wystąpienia litery a

5. *Znajdź wszystkie wystąpienia słowa "p a t t e r n" w haystack.png.*

   Należy zaznaczyć że współrzędne wystąpienia wzorca mogą nieznacznie odchylać się od rzeczywistych ze względu na niedokładności przy wycinaniu zdjęcia.

```
Pattern appeared:  [(391, 183), (413, 427), (457, 241), (501, 141), (545, 247)]
```

Rysunek 6 Zad5 Wyniki

6. *Porównaj czas budowania automatu i czas wyszukiwania dla różnych rozmiarów wzorca*

```
=== Small text ===
Text finding
Building automata took: 0.0 s
Finding took: 0.02648162841796875 s
Image finding
Building automata took: 0.0 s
Finding took: 3.608320951461792 s

=== Medium text ===
Text finding
Building automata took: 0.0 s
Finding took: 0.01080465316772461 s
Image finding
Building automata took: 0.021935701370239258 s
Finding took: 3.847768783569336 s

=== Big text ===
Text finding
Building automata took: 0.0 s
Finding took: 0.01109170913696289 s
Image finding
Building automata took: 0.15472745895385742 s
Finding took: 4.072037935256958 s

=== Large text ===
Text finding
Building automata took: 0.0 s
Finding took: 0.021561861038208008 s
Image finding
Building automata took: 1.0515666007995605 s
Finding took: 2.9891037940979004 s
```

Rysunek 7 Zad6 wyniki

7. Podziel plik na 2, 4 i 8 fragmentów (w poziomie) i porównaj czas przeszukiwania

```
1 parts took 0.029915332794189453 s
2 parts took 0.019945621490478516 s
4 parts took 0.012928962707519531 s
8 parts took 0.012041330337524414 s
```

Rysunek 8 Zad7 Wyniki

8. Wnioski
   - Sprawdzając powyższe testy z rzeczywistymi wystąpieniami wzorców w plikach, możemy stwierdzić, że zaimplementowany algorytm działa prawidłowo.
   - Widzimy, że czas działania algorytmu Aho-Corasick w naszych wynikach nie wzrasta jednoznacznie wraz ze wzrostem rozmiaru wzorca. Dzieje się tak ponieważ złożoność algorytmu jest liniowo zależna od sumy długości wzorców, długości tekstu, ale także liczby wystąpień wzorców w tekście. Dlatego też, większy jest czas wyszukiwania dla wzorca najmniejszego niż największego.
   - Czasy wyszukiwania wzorca w zdjęciach są bardziej czasochłonne.

- Budowa automatu skończonego jest liniowo zależna od długości wzorca. Natomiast przez niedokładności naszego algorytmu budowy automatu nie możemy tego potwierdzić wynikami.
- Możliwym jest, że podzielenie piku na fragmenty powoduje zmniejszenie całkowitego czasu wyszukiwania. Ciężko podać powód takiego przyśpieszenia i stwierdzić jego słuszność, tym bardziej, że kolejne pomiary różniły się od siebie znacząco.