**TUGAS KECIL IV**

**IF2211 STRATEGI ALGORITMA**

**EKSTRAKSI INFORMASI DARI ARTIKEL BERITA DENGAN ALGORITMA PENCOCOKAN STRING**

**LAPORAN TUGAS KECIL**

Diajukan untuk memenuhi tugas mata kuliah IF2211 Strategi Algoritma

Oleh

**MUHAMMAD FAUZAN AL-GHIFARI**

**(13518112 - K01)**

****

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2020**

1. **Deskripsi Singkat Pncocokan String KMP**

Algoritma Knuth-Morris-Pratt merupakan salah satu algoritma pencarian string (string matcing) yang dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977 ([Wikipedia : Knuth-Morris-Pratt](http://id.wikipedia.org/wiki/Algoritma_Knuth-Morris-Pratt)). Langkah-langkah yang dilakukan algoritma Knuth-Morris-Pratt pada saat mencocokkan string yaitu (modifikasi):

String pattern (kata yang dicari) akan dipecah menjadi array karakter String text (teks, artikel, dsb) akan dipecah menjadi array karakter Menentukan lompatan yang akan dilakukan ketika pencarian (funsi preKMP()) Algoritma Knuth-Morris-Pratt mulai mencocokkan pattern pada awal teks. ()

Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi: Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.

Algoritma kemudian menggeser pattern berdasarkan tabel next (lompat), lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

Cuplikan program bagian algoritma KMP buatan saya

|  |
| --- |
| from copy import deepcopy  *# Knuth Morris Pratt Algorithm*  def KMP (T, P):      bf = borderFunction(P)      result = []      n = len(T)      m = len(P)      i = 0 *# for teks*      j = 0 *# for pattern*      temp = j      while (i < n):          while (j < m and i < n):              if (j == -99):                  j = temp+1                  i += 1              elif (j == m-1 and P[j] == T[i]):                  result.append(i-m+1)                  i += 1                  j = 0              elif (T[i] == P[j]):                  i += 1;                  j += 1;              else: *# miss match*                  temp = j                  j = bf[j]          i+=1      return result  *# Make Border Function*  def borderFunction(pattern):      j = len(pattern);      bf = []      for i in range (j):          bf.append(calculateBF(pattern, i))          bf[0] = -99      return bf  *# Calculate BF for specific j*  def calculateBF(pattern, j):      ret = 0      stop = False      pref = []      suf = []  *# calculate pref*      input = []      for i in range (0, j, 1):          input.append(pattern[i])          pref.append(deepcopy(input))  *# calculate suf*      input.clear()      for i in range (j-1, 0, -1):          input.insert(0,pattern[i])          suf.append(deepcopy(input))  *# looking for b[k]*      i = len(suf)-1      while i >= 0 and not(stop):          if (pref[i] == suf[i]):              ret = len(pref[i])              stop = True          i -= 1      return ret  from nltk.tokenize import sent\_tokenize  def main():      f = open ("text.txt", "r")      T = f.read()  *# P = "COVID-19"*  *# T = "saya adalah fauzan keren, memang fauzan keren, sudah tentu fauzan keren"*  *# P = "fauzan"*  *# print(KMP(T,P))*    if \_\_name\_\_ == "\_\_main\_\_":      main() |

1. **Deskripsi Pencocokan String dengan Boyeer Moore**

Algoritma Boyer-Moore diperkenalkan oleh Bob Boyer dan J.S. Moore pada tahun 1977. Pada algoritma ini pencocokan kata dimulai dari karakter terakhir kata kunci menuju karakter awalnya. Jika terjadi perbedaan antara karakter terakhir kata kunci dengan kata yang dicocokkan maka karakter-karakter dalam potongan kata yang dicocokkan tadi akan diperiksa satu per satu. Hal ini dimaksudkan untuk mendeteksi apakah ada karakter dalam potongan kata tersebut yang sama dengan karakter yang ada pada kata kunci. Apabila terdapat kesamaan, maka kata kunci akan digeser sedemikian rupa sehingga posisi karakter yang sama terletak sejajar, dan kemudian dilakukan kembali pencocokan karakter terakhir dari kata kunci. Sebaliknya jika tidak terdapat kesamaan karakter, maka seluruh karakter kata kunci akan bergeser ke kanan sebanyak m karakter, di mana m adalah panjang karakter dari kata kunci. Booyer-Moore merupakan salah satu Algortima Pattern Matching yang cukup terkenal. Algoritma ini menggunakan beberapa kasus pengecekan teks (input karakter yang akan dibaca) dengan Pattern (pola yang akan disaring). Algoritma Boyer-Moore adalah algoritma pencarian string yang mencari dengan cara membandingkan sebuah huruf dengan huruf yang ada di pattern yang dicari, dan menggeser pattern tersebut hingga posisinya sama dengan teks yang dicari dan membandingkan kata tersebut. Cara ini disebut character jump

Contoh Algoritma Boyer Moore buatan saya

1. **Screenshot input-output program**

Ubin kosong direpresentasikan dengan ` ` spasi kosong

1. File test1.txt

|  |
| --- |
| 2 3 4 8  1 7 11  5 9 6 12  13 10 14 15 |

output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 1  Kondisi Awal Puzzle  -------------  | 2| 3| 4| 8|  | 1| 7|11|  |  | 5| 9| 6|12|  |13|10|14|15|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           1     3           1     4           1     5           0     6           0     7           2     8           4     9           1     10          0     11          4     12          1     13          1     14          0     15          0     16          8  -------------------------  X = 0  Sigma (KURANG(i)) + X = 24  Jumlah simpul yang dibangkitkan = 47  Urutan penyelesaian = ['U', 'L', 'L', 'L', 'D', 'D', 'R', 'R', 'U', 'L', 'D', 'D', 'R', 'R']  Step ke 1 : U  -------------  | 2| 3| 4|  |  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 2 : L  -------------  | 2| 3|  | 4|  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 3 : L  -------------  | 2|  | 3| 4|  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 4 : L  -------------  |  | 2| 3| 4|  | 1| 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 5 : D  -------------  | 1| 2| 3| 4|  |  | 7|11| 8|  | 5| 9| 6|12|  |13|10|14|15|  -------------  Step ke 6 : D  -------------  | 1| 2| 3| 4|  | 5| 7|11| 8|  |  | 9| 6|12|  |13|10|14|15|  -------------  Step ke 7 : R  -------------  | 1| 2| 3| 4|  | 5| 7|11| 8|  | 9|  | 6|12|  |13|10|14|15|  -------------  Step ke 8 : R  -------------  | 1| 2| 3| 4|  | 5| 7|11| 8|  | 9| 6|  |12|  |13|10|14|15|  -------------  Step ke 9 : U  -------------  | 1| 2| 3| 4|  | 5| 7|  | 8|  | 9| 6|11|12|  |13|10|14|15|  -------------  Step ke 10 : L  -------------  | 1| 2| 3| 4|  | 5|  | 7| 8|  | 9| 6|11|12|  |13|10|14|15|  -------------  Step ke 11 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|  |11|12|  |13|10|14|15|  -------------  Step ke 12 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|  |14|15|  -------------  Step ke 13 : R  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|  |15|  -------------  Step ke 14 : R  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|15|  |  -------------  Waktu eksekusi = 0.10163068771362305 seconds |

1. File test2.txt

|  |
| --- |
| 2 14 10 4  1 7 3 8  5 6 12  9 13 11 15 |

Output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 2  Kondisi Awal Puzzle  -------------  | 2|14|10| 4|  | 1| 7| 3| 8|  |  | 5| 6|12|  | 9|13|11|15|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           1     3           0     4           2     5           0     6           0     7           3     8           2     9           0     10          8     11          0     12          2     13          1     14          12     15          0     16          7  -------------------------  X = 0  Sigma (KURANG(i)) + X = 38  Jumlah simpul yang dibangkitkan = 4063  Urutan penyelesaian = ['R', 'U', 'U', 'R', 'D', 'D', 'L', 'U', 'U', 'L', 'D', 'D', 'D', 'R', 'U', 'U', 'R', 'D', 'D', 'R']  Step ke 1 : R  -------------  | 2|14|10| 4|  | 1| 7| 3| 8|  | 5|  | 6|12|  | 9|13|11|15|  -------------  Step ke 2 : U  -------------  | 2|14|10| 4|  | 1|  | 3| 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 3 : U  -------------  | 2|  |10| 4|  | 1|14| 3| 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 4 : R  -------------  | 2|10|  | 4|  | 1|14| 3| 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 5 : D  -------------  | 2|10| 3| 4|  | 1|14|  | 8|  | 5| 7| 6|12|  | 9|13|11|15|  -------------  Step ke 6 : D  -------------  | 2|10| 3| 4|  | 1|14| 6| 8|  | 5| 7|  |12|  | 9|13|11|15|  -------------  Step ke 7 : L  -------------  | 2|10| 3| 4|  | 1|14| 6| 8|  | 5|  | 7|12|  | 9|13|11|15|  -------------  Step ke 8 : U  -------------  | 2|10| 3| 4|  | 1|  | 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 9 : U  -------------  | 2|  | 3| 4|  | 1|10| 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 10 : L  -------------  |  | 2| 3| 4|  | 1|10| 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 11 : D  -------------  | 1| 2| 3| 4|  |  |10| 6| 8|  | 5|14| 7|12|  | 9|13|11|15|  -------------  Step ke 12 : D  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  |  |14| 7|12|  | 9|13|11|15|  -------------  Step ke 13 : D  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  | 9|14| 7|12|  |  |13|11|15|  -------------  Step ke 14 : R  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  | 9|14| 7|12|  |13|  |11|15|  -------------  Step ke 15 : U  -------------  | 1| 2| 3| 4|  | 5|10| 6| 8|  | 9|  | 7|12|  |13|14|11|15|  -------------  Step ke 16 : U  -------------  | 1| 2| 3| 4|  | 5|  | 6| 8|  | 9|10| 7|12|  |13|14|11|15|  -------------  Step ke 17 : R  -------------  | 1| 2| 3| 4|  | 5| 6|  | 8|  | 9|10| 7|12|  |13|14|11|15|  -------------  Step ke 18 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|  |12|  |13|14|11|15|  -------------  Step ke 19 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|  |15|  -------------  Step ke 20 : R  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|15|  |  -------------  Waktu eksekusi = 0.6677596569061279 seconds |

1. File test3.txt

|  |
| --- |
| 1 2 7 3  11 9 10 6  12 14 4  5 13 15 8 |

Output

|  |
| --- |
| A  pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 3  Kondisi Awal Puzzle  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  |12|  |14| 4|  | 5|13|15| 8|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           0     3           0     4           0     5           0     6           2     7           4     8           0     9           4     10          4     11          6     12          3     13          1     14          4     15          1     16          6  -------------------------  X = 1  Sigma (KURANG(i)) + X = 36  Jumlah simpul yang dibangkitkan = 212534  Urutan penyelesaian = ['L', 'D', 'R', 'U', 'U', 'L', 'D', 'R', 'R', 'D', 'L', 'U', 'U', 'R', 'R', 'D', 'D', 'L', 'U', 'L', 'U', 'R', 'U', 'R', 'D', 'D', 'D']  Step ke 1 : L  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  |  |12|14| 4|  | 5|13|15| 8|  -------------  Step ke 2 : D  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  | 5|12|14| 4|  |  |13|15| 8|  -------------  Step ke 3 : R  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  | 5|12|14| 4|  |13|  |15| 8|  -------------  Step ke 4 : U  -------------  | 1| 2| 7| 3|  |11| 9|10| 6|  | 5|  |14| 4|  |13|12|15| 8|  -------------  Step ke 5 : U  -------------  | 1| 2| 7| 3|  |11|  |10| 6|  | 5| 9|14| 4|  |13|12|15| 8|  -------------  Step ke 6 : L  -------------  | 1| 2| 7| 3|  |  |11|10| 6|  | 5| 9|14| 4|  |13|12|15| 8|  -------------  Step ke 7 : D  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  |  | 9|14| 4|  |13|12|15| 8|  -------------  Step ke 8 : R  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|  |14| 4|  |13|12|15| 8|  -------------  Step ke 9 : R  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|14|  | 4|  |13|12|15| 8|  -------------  Step ke 10 : D  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|14|15| 4|  |13|12|  | 8|  -------------  Step ke 11 : L  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|14|15| 4|  |13|  |12| 8|  -------------  Step ke 12 : U  -------------  | 1| 2| 7| 3|  | 5|11|10| 6|  | 9|  |15| 4|  |13|14|12| 8|  -------------  Step ke 13 : U  -------------  | 1| 2| 7| 3|  | 5|  |10| 6|  | 9|11|15| 4|  |13|14|12| 8|  -------------  Step ke 14 : R  -------------  | 1| 2| 7| 3|  | 5|10|  | 6|  | 9|11|15| 4|  |13|14|12| 8|  -------------  Step ke 15 : R  -------------  | 1| 2| 7| 3|  | 5|10| 6|  |  | 9|11|15| 4|  |13|14|12| 8|  -------------  Step ke 16 : D  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|15|  |  |13|14|12| 8|  -------------  Step ke 17 : D  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|15| 8|  |13|14|12|  |  -------------  Step ke 18 : L  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|15| 8|  |13|14|  |12|  -------------  Step ke 19 : U  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|11|  | 8|  |13|14|15|12|  -------------  Step ke 20 : L  -------------  | 1| 2| 7| 3|  | 5|10| 6| 4|  | 9|  |11| 8|  |13|14|15|12|  -------------  Step ke 21 : U  -------------  | 1| 2| 7| 3|  | 5|  | 6| 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 22 : R  -------------  | 1| 2| 7| 3|  | 5| 6|  | 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 23 : U  -------------  | 1| 2|  | 3|  | 5| 6| 7| 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 24 : R  -------------  | 1| 2| 3|  |  | 5| 6| 7| 4|  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 25 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7|  |  | 9|10|11| 8|  |13|14|15|12|  -------------  Step ke 26 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|  |  |13|14|15|12|  -------------  Step ke 27 : D  -------------  | 1| 2| 3| 4|  | 5| 6| 7| 8|  | 9|10|11|12|  |13|14|15|  |  ------------- |

1. File test4.txt

|  |
| --- |
| 3 9 1 15  14 11 4 6  13 10 12  2 7 8 5 |

Output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 4  Kondisi Awal Puzzle  -------------  | 3| 9| 1|15|  |14|11| 4| 6|  |13|  |10|12|  | 2| 7| 8| 5|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           0     3           2     4           1     5           0     6           2     7           1     8           1     9           7     10          4     11          7     12          4     13          6     14          10     15          11     16          6  -------------------------  X = 1  Sigma (KURANG(i)) + X = 63  Permasalahan Puzzle tidak dapat diselesaikan  Waktu eksekusi = 0.0 seconds |

1. File test5.txt

|  |
| --- |
| 1 3 4  15 2 13 9  10 11 12 6  7 8 5 14 |

Output

|  |
| --- |
| pilih file test  1. test1.txt  2. test2.txt  3. test3.txt  4. test4.txt  5. test5.txt  input angka 1-5 : 5  Kondisi Awal Puzzle  -------------  | 1| 3|  | 4|  |15| 2|13| 9|  |10|11|12| 6|  | 7| 8| 5|14|  -------------  Fungsi Kurang untuk setiap ubin  -------------------------  Ubin-i   FungsiKurang(i)     1           0     2           0     3           1     4           1     5           0     6           1     7           1     8           1     9           4     10          4     11          4     12          4     13          8     14          0     15          11     16          13  -------------------------  X = 0  Sigma (KURANG(i)) + X = 53  Permasalahan Puzzle tidak dapat diselesaikan  Waktu eksekusi = 0.0 seconds |

**DISCLAIMER**

Secara teori program ini dapat menyelesaikan semua permasalahan 15 Puzzle karena saya sudah mencocokkan hasilnya dengan pekerjakan manual yang saya kerjakan berdasar pada test1.txt, tetapi untuk beberapa kasus matriks yang dibangkitkan terlalu banyak sehingga membutuhkan waktu yang sangat lama atau sebelum mencapai solusi terjadi memory error. Jika ada test case yang tidak berhasil diselesaikan oleh program ini tetapi bisa diselesaikan oleh program lain itu disebabkan oleh pemilihan simpul dengan bobot terkecil saat bobot terkecil jumlahnya lebih dari satu (pemilihan simpul yang diekspan dibebaskan), saya sendiri memilih untuk mengekspan simpul yang paling baru jika ada beberapa simpul dengan bobot yang sama.

**Spesifikasi komputer yang digunakan**

Nama Laptop : Lenovo IdeaPad S340

OS : Windows 10

RAM : 8192 MB / 8 GB

Processor : AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx (4CPUs), ~~2.6GHz

|  |  |  |  |
| --- | --- | --- | --- |
| No | Point | Ya | Tidak |
| 1 | Program berhasil dikompilasi | ✓ |  |
| 2 | Program berhasil *running* | ✓ |  |
| 3 | Program dapat menerima input dan menuliskan output | ✓ |  |
| 4 | Luaran sudah benar untuk semua data uji | ✓ |  |