

Tugas Kecil 3 IF2211 Strategi Algoritma

Penyelesaian Persoalan 15-Puzzle dengan Algoritma *Branch and Bound*



Nama : Ahmad Romy Zahran

NIM : 13520009

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2022

A. Algoritma *Branch and Bound*

Program menggunakan algoritma *Branch and Bound* dengan langkah-langkah sebagai berikut.

1. Program menerima opsi posisi awal dari file atau secara random. Bila dipilih posisi awal dari file, program menerima path input file. Bila dipilih posisi awal random, program membuat puzzle secara random.
2. Program mengecek apakah posisi awal dapat diselesaikan dengan menghitung sigma $KURANG(i) + X$ genap atau tidak. Bila tidak, program mencetak 15-puzzle tidak dapat diselesaikan dan program selesai.
 - a) $KURANG(i)$ = banyak bilangan $j < i$ dengan $posisi(j) > posisi(i)$, untuk kotak kosong atau “-“ dianggap sebagai angka 16. Dengan prinsip komplemen, $Kurang(i)$ sama saja dengan $(i-1) - (\text{banyak } j < i \text{ dengan } posisi(j) < posisi(i))$.
 - b) $X = 1$ bila kotak kosong ada di posisi dengan nomor baris + nomor kolom bernilai ganjil.
3. Program menyiapkan Priority Queue pq , mp sebagai dictionary/map yang menyimpan *best $f(p)$ so far* untuk setiap posisi puzzle p yang tercapai, $mpAction$ sebagai aksi yang digunakan untuk mencapai posisi tertentu dan $ctNode$ sebagai banyak simpul yang dibangkitkan. Timer eksekusi dimulai.
4. Program menjalankan algoritma *Branch and Bound*.
 - a) Program menghitung $f(p)$, $\hat{g}(p)$, $\hat{c}(p)$ dari simpul akar serta memasukkannya bersama simpul akar ke antrian pq sebagai 1 elemen. $ctNode$ ditambah 1.
 - $f(p)$ adalah banyak aksi untuk mencapai posisi p dari posisi awal. $f(p)$ untuk posisi awal adalah nol.
 - $\hat{g}(p)$ adalah aproksimasi banyak aksi untuk mencapai simpul solusi dan bertindak sebagai batas bawah. Digunakan aproksimasi berupa banyak kotak tidak kosong yang belum berada di posisi solusi.
 - $\hat{c}(p) = f(p) + \hat{g}(p)$. Bertindak sebagai penentu arah pembangkitan ruang solusi yang dibangkitkan. Simpul terdepan pada antrian memiliki $\hat{c}(p)$ terkecil.
 - b) Selama pq tidak kosong, program mengambil simpul terdepan, misal simpul u . Karena posisi awal dapat diselesaikan, pq tidak akan kosong sebelum ditemukan solusi.
 - c) Bila $f(u)$ lebih besar dari *best $f(p)$ so far* dengan posisi yang sama yang disimpan mp , abaikan simpul u dan kembali ke langkah b).
 - d) Bila u sudah berposisi solusi, maka $f(u)$ adalah solusi dan dapat lanjut ke langkah 5.
 - e) Untuk setiap aksi (up, right, left, down) yang dapat dilakukan pada posisi u , lakukan aksi tersebut pada simpul u menghasilkan posisi v dengan $f(v) = f(u)+1$. Bila $f(v)$ belum tercatat di mp atau lebih kecil dari *best $f(v)$ so far*, perbarui *best $f(v)$ so far* dan masukkan posisi v beserta $f(v)$ dan perhitungan $\hat{g}(v)$ dan $\hat{c}(v)$ ke antrian pq sebagai 1 elemen serta

tambahkan ctNode dan catat aksi yang menghasilkan v. *best f(p) so far* bertindak sebagai bound agar posisi v dengan path yang tidak efisien tidak dimasukkan ke ruang solusi. Selanjutnya kembali ke langkah b).

5. Hentikan timer, dari posisi solusi gunakan mpAction untuk mengetahui aksi sebelumnya dan lakukan kebalikan aksi sebelumnya (up dibalas down, left dibalas right) untuk mendapat posisi sebelumnya. Telusuri hingga posisi awal dan output posisi awal, beserta urutan aksi dan hasil aksi hingga posisi solusi. Selanjutnya output banyak langkah yang dibutuhkan, waktu eksekusi, dan jumlah simpul yang dibangkitkan.

B. Source Code Program

Source code program menggunakan bahasa python dan terdiri dari file 15-puzzle-solver.py sebagai program utama dan deklarasi kelas elemen di Priority Queue serta file util.py yang menyediakan variabel dan fungsi pembantu.

1. 15-puzzle-solver.py

```
1 import time
2 from queue import PriorityQueue
3 from util import *
4
5 # wrapper puzzle yang memakai c(p) sebagai prioritas
6 class PuzzleBnB:
7     def __init__(self,fp,gp,puzzle):
8         self.fp = fp
9         self.gp = gp
10        self.priority = fp+gp
11        self.puzzle = puzzle
12
13    def __lt__(self, other):
14        return self.priority < other.priority
15
16 # membuat puzzle
17 while(True):
18     print("Pilih opsi pembangkitan posisi awal 15-Puzzle")
19     print("1. Dari file")
20     print("2. Dari random generator")
21     opsi = int(input("Masukkan nomor opsi: "))
22     if(opsi==1):
23         puzzle = inputPuzzleFromFile()
24         break
25     elif(opsi==2):
26         puzzle = randomPuzzle()
27         break
28     else:
29         print("Opsii tidak valid")
30
31 printDivider()
32 print("Matriks posisi awal 15-puzzle:")
33 printPuzzle(puzzle)
34 sum = printKURANG_I_X(puzzle)
35 printDivider()
36 if(sum%2==1):
37     print("15-puzzle tidak dapat diselesaikan karena hasil penjumlahan bernilai ganjil")
38 else:
39     print("15-puzzle dapat diselesaikan!")
40     print("MULAI PENCARIAN...")
```

```

39     print("15-puzzle dapat diselesaikan!")
40     print("MULAI  PENCARIAN...")
41     pq = PriorityQueue()
42     mp = {puzzle:0} # dict/map penyimpan best fp so far
43     mpAction = dict()
44     ctNode = 0
45     start = time.time()
46     pq.put(PuzzleBnB(0,g(puzzle),puzzle))
47     ctNode +=1
48     while(pq.qsize()>0):
49         PrioPuzzle = pq.get()
50         fpu = PrioPuzzle.fp
51         pu = PrioPuzzle.puzzle
52         if(fpu>mp[pu]):
53             continue
54         if(sol(pu)):
55             final = pu
56             ctStep = fpu
57             break
58         for k in range(4):
59             if(canMove(k,pu)):
60                 pv = after(k,pu)
61                 if((pv not in mp) or mp[pv]>fpu+1):
62                     mp[pv] = fpu+1
63                     mpAction[pv] = k
64                     pq.put(PuzzleBnB(fpu+1,g(pv),pv))
65                     ctNode +=1
66
67     # pencarian selesai
68     end = time.time()
69     elapsedTime = end-start
70     print("PENCARIAN  SELESAI")
71     printDivider()
72
73     list_of_puzzle = []
74     u = final
75     while(u!=puzzle):
76         k = mpAction[u]
77         list_of_puzzle.append((u,action[k]))
78         u = before(k,u)
79     list_of_puzzle.reverse()
80     print("Posisi awal:")
81     printPuzzle(puzzle)
82     printDivider()
83     ctAksi=1
84     for u, act in list_of_puzzle:
85         print("Aksi ke-{}: {}".format(ctAksi)+act)
86         print("Hasil setelah aksi:")
87         printPuzzle(u)
88         printDivider()
89         ctAksi +=1
90
91     print("Langkah yang dibutuhkan: {}".format(ctStep))
92     print("Waktu eksekusi program: {}".format(elapsedTime))
93     print("Jumlah simpul yang dibangkitkan: {}".format(ctNode))
94

```

2. util.py

```

1  import random
2
3  # VAR GLOBAL
4  dr = [1,0,-1,0]
5  dc = [0,-1,0,1]
6  action = ["up","right","down","left"]
7

```

```

8 def randomPuzzle():
9     puzzle = ["1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","-"]
10    random.shuffle(puzzle)
11    # print(tuple(puzzle))
12    return tuple(puzzle)
13
14 def inputPuzzleFromFile():
15     while(True):
16         pathInput = input("Masukkan path file input: ")
17         try:
18             file = open(pathInput, "r")
19             puzzle = tuple(file.read().split())
20             # print(puzzle)
21             return puzzle
22         except(EOFError):
23             print("path input tidak valid")
24
25 def printDivider():
26     print("=====")
27
28 def printPuzzle(puzzle):
29     ct = 0
30     for i in range(4):
31         for j in range(4):
32             print(puzzle[ct], end=" ")
33             ct +=1
34         print()
35
36 def KURANG(i, puzzle):
37     kecil_I = 0
38     if(i==16):
39         for k in range(16):
40             if(puzzle[k]=="-"):
41                 break
42             elif(int(puzzle[k])<i):
43                 kecil_I +=1
44     else:
45         for k in range(16):
46             if(puzzle[k]=="-"):
47                 continue # 16>i
48             elif(int(puzzle[k])==i):
49                 break
50             elif(int(puzzle[k])<i):
51                 kecil_I +=1
52     return (i-1-kecil_I)
53
54 # jika "-" di baris+kolom ganjil return 1, else return 0
55 def X(puzzle):
56     idx = puzzle.index("-")
57     # print("idx "+str(idx))
58     baris = idx //4
59     kolom = idx % 4
60     if((baris+kolom)%2==1):
61         return 1
62     else:
63         return 0
64
65 # mencetak nilai KURANG(i),X dan mengembalikan totalnya
66 def printKURANG_I_X(puzzle):
67     sum = 0
68     print("Berikut nilai KURANG(i) dan X:")
69     for i in range(1,17):
70         kurangi = KURANG(i,puzzle)
71         print("KURANG({}) = {}".format(i,KURANG(i,puzzle)))
72         sum += kurangi
73     x = X(puzzle)
74     print("X= {}".format(x))
75     sum += x
76     print("SIGMA KURANG(i) + X = {}".format(sum))
77     return sum

```

```

79 def g(puzzle):
80     ct = 0
81     for i in range(16):
82         if(puzzle[i]!="-" and int(puzzle[i])!=i+1):
83             ct +=1
84     return ct
85
86 def sol(puzzle):
87     for i in range(15):
88         if(puzzle[i]!="-" or int(puzzle[i])!=i+1):
89             return False
90     if(puzzle[15]!="-"):
91         return False
92     return True
93
94 def toIdx(r,c):
95     return 4*r+c
96
97 def toRC(idx):
98     return idx//4,idx%4
99
100 # cek puzzle dapat melakukan aksi ke-k
101 def canMove(k,puzzle):
102     r,c = toRC(puzzle.index("-"))
103     rk = r+dr[k]; ck = c+dc[k]
104     return (rk>=0 and rk<=3 and ck>=0 and ck<=3)
105
106 # mengembalikan posisi puzzle setelah aksi ke-k
107 def after(k,puzzle):
108     r,c = toRC(puzzle.index("-"))
109     rk = r+dr[k]; ck = c+dc[k]
110     p = list(puzzle)
111     idx = toIdx(r,c); idxk = toIdx(rk,ck)
112     temp = p[idx]
113     p[idx] = p[idxk]
114     p[idxk] = temp
115     return tuple(p)
116
117 #mengembalikan posisi puzzle sebelum melakukan aksi ke-k
118 def before(k,puzzle):
119     return after((k+2)%4,puzzle)

```

C. Tangkapan Layar Hasil Pengujian

Pada folder test terdapat 2 kasus tidak dapat diselesaikan berupa unsolved_<nomor>.in dan 3 kasus dapat diselesaikan bernama solved_<nomor>.in. Berikut isinya:

```

test > ≡ unsolved_1.in
1  1  3  4 15
2  2  -  5 12
3  7  6 11 14
4  8  9 10 13
5  |




```

```

test > ≡ unsolved_2.in
1  2 13 11 3
2 14 5 9 4
3  6  7 - 12
4 10 1 8 15
5  |

```

Testcase Unsolved

test >  solved_1.in	test >  solved_2.in	test >  solved_3.in
1 3 2 4 8	1 1 3 4 8	1 1 3 4 8
2 9 5 7 6	2 5 2 7 6	2 9 14 11 -
3 1 - 10 11	3 9 10 - 11	3 13 7 5 12
4 13 14 15 12	4 13 14 15 12	4 6 2 15 10
5	5	5

Testcase Solved

Pertama, program meminta input nomor opsi hingga valid (1 bila posisi awal diambil dari file dan 2 melalui random generator). Bila diambil dari file, program meminta path file hingga valid. Berikut contoh input-output program.

```
PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 3
Opsi tidak valid
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 2
=====
Matriks posisi awal 15-puzzle:
6 9 12 2
1 10 5 15
13 - 8 14
7 11 3 4
Berikut nilai KURANG(i) dan X:
KURANG(1) = 0
KURANG(2) = 1
KURANG(3) = 0
KURANG(4) = 0
KURANG(5) = 2
KURANG(6) = 5
KURANG(7) = 2
KURANG(8) = 3
KURANG(9) = 7
KURANG(10) = 5
KURANG(11) = 2
KURANG(12) = 9
KURANG(13) = 5
KURANG(14) = 4
KURANG(15) = 7
KURANG(16) = 6
X= 1
SIGMA KURANG(i) + X = 59
=====
15-puzzle tidak dapat diselesaikan karena hasil penjumlahan bernilai ganjil
PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009>
```

Contoh input opsi tidak valid dan input output dengan puzzle random

```
PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 1
Masukkan path file input: test/asal.in
path input tidak valid
Masukkan path file input: 
```

Contoh ketika path input tidak valid

Berikutnya ada contoh input-output program dari 5 berkas yang sudah disiapkan pada folder test.

1. unsolved_1.in

```
PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 1
Masukkan path file input: test/unsolved_1.in
=====
```

input

```
Matriks posisi awal 15-puzzle:
1 3 4 15
2 - 5 12
7 6 11 14
8 9 10 13
```

Berikut nilai KURANG(i) dan X:

```
KURANG(1) = 0
KURANG(2) = 0
KURANG(3) = 1
KURANG(4) = 1
KURANG(5) = 0
KURANG(6) = 0
KURANG(7) = 1
KURANG(8) = 0
KURANG(9) = 0
KURANG(10) = 0
KURANG(11) = 3
KURANG(12) = 6
KURANG(13) = 0
KURANG(14) = 4
KURANG(15) = 11
KURANG(16) = 10
X= 0
```

```
SIGMA KURANG(i) + X = 37
=====
```

```
15-puzzle tidak dapat diselesaikan karena hasil penjumlahan bernilai ganjil
```

output

2. unsolved_2.in


```

PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 1
Masukkan path file input: test/unsolved_2.in
=====

```

input

```

Matriks posisi awal 15-puzzle:
2 13 11 3
14 5 9 4
6 7 - 12
10 1 8 15
Berikut nilai KURANG(i) dan X:
KURANG(1) = 0
KURANG(2) = 1
KURANG(3) = 1
KURANG(4) = 1
KURANG(5) = 2
KURANG(6) = 1
KURANG(7) = 1
KURANG(8) = 0
KURANG(9) = 5
KURANG(10) = 2
KURANG(11) = 9
KURANG(12) = 3
KURANG(13) = 11
KURANG(14) = 9
KURANG(15) = 0
KURANG(16) = 5
X= 0
SIGMA KURANG(i) + X = 51
=====
15-puzzle tidak dapat diselesaikan karena hasil penjumlahan bernilai ganjil
=====

```

output

3. solved_1.in. Membutuhkan 21 langkah.

```

PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 1
Masukkan path file input: test/solved_1.in

```

input

```

Matriks posisi awal 15-puzzle:
3 2 4 8
9 5 7 6
1 - 10 11
13 14 15 12
Berikut nilai KURANG(i) dan X:
KURANG(1) = 0
KURANG(2) = 1
KURANG(3) = 2
KURANG(4) = 1
KURANG(5) = 1
KURANG(6) = 1
KURANG(7) = 2
KURANG(8) = 4
KURANG(9) = 4
KURANG(10) = 0
KURANG(11) = 0
KURANG(12) = 0
KURANG(13) = 1
KURANG(14) = 1
KURANG(15) = 1
KURANG(16) = 6
X= 1
SIGMA KURANG(i) + X = 26
=====
15-puzzle dapat diselesaikan!
MULAI PENCARIAN...
PENCARIAN SELESAI
=====
Posisi awal:
3 2 4 8
9 5 7 6
1 - 10 11
13 14 15 12
=====
Aksi ke-1: down
Hasil setelah aksi:
3 2 4 8
9 - 7 6
1 5 10 11
13 14 15 12
=====
Aksi ke-2: right
Hasil setelah aksi:
3 2 4 8
- 9 7 6
1 5 10 11
13 14 15 12
=====

```

```

Aksi ke-3: up
Hasil setelah aksi:
3 2 4 8
1 9 7 6
- 5 10 11
13 14 15 12
=====
Aksi ke-4: left
Hasil setelah aksi:
3 2 4 8
1 9 7 6
5 - 10 11
13 14 15 12
=====
Aksi ke-5: down
Hasil setelah aksi:
3 2 4 8
1 - 7 6
5 9 10 11
13 14 15 12
=====
Aksi ke-6: down
Hasil setelah aksi:
3 - 4 8
1 2 7 6
5 9 10 11
13 14 15 12
=====
Aksi ke-7: right
Hasil setelah aksi:
- 3 4 8
1 2 7 6
5 9 10 11
13 14 15 12
=====
Aksi ke-8: up
Hasil setelah aksi:
1 3 4 8
- 2 7 6
5 9 10 11
13 14 15 12
=====
Aksi ke-9: up
Hasil setelah aksi:
1 3 4 8
5 2 7 6
- 9 10 11
13 14 15 12
=====

```

```

Aksi ke-10: left
Hasil setelah aksi:
1 3 4 8
5 2 7 6
9 - 10 11
13 14 15 12
=====
Aksi ke-11: left
Hasil setelah aksi:
1 3 4 8
5 2 7 6
9 10 - 11
13 14 15 12
=====
Aksi ke-12: down
Hasil setelah aksi:
1 3 4 8
5 2 - 6
9 10 7 11
13 14 15 12
=====
Aksi ke-13: left
Hasil setelah aksi:
1 3 4 8
5 2 6 -
9 10 7 11
13 14 15 12
=====
Aksi ke-14: down
Hasil setelah aksi:
1 3 4 -
5 2 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-15: right
Hasil setelah aksi:
1 3 - 4
5 2 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-16: right
Hasil setelah aksi:
1 - 3 4
5 2 6 8
9 10 7 11
13 14 15 12

```

```

Aksi ke-17: up
Hasil setelah aksi:
1 2 3 4
5 - 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-18: left
Hasil setelah aksi:
1 2 3 4
5 6 - 8
9 10 7 11
13 14 15 12
=====
Aksi ke-19: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 - 11
13 14 15 12
=====
Aksi ke-20: left
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 11 -
13 14 15 12
=====
Aksi ke-21: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 -
=====
Langkah yang dibutuhkan: 21
Waktu eksekusi program: 0.14302968978881836
Jumlah simpul yang dibangkitkan: 13422
D5-D:\ITD\Sem-4\Struc\Tug11\Tug11-13520000

```

output

4. solved_2.in. Membutuhkan 10 langkah.

```
PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 1
Masukkan path file input: test/solved_2.in
=====
```

input

```

Matriks posisi awal 15-puzzle:
1 3 4 8
5 2 7 6
9 10 - 11
13 14 15 12
Berikut nilai KURANG(i) dan X:
KURANG(1) = 0
KURANG(2) = 0
KURANG(3) = 1
KURANG(4) = 1
KURANG(5) = 1
KURANG(6) = 0
KURANG(7) = 1
KURANG(8) = 4
KURANG(9) = 0
KURANG(10) = 0
KURANG(11) = 0
KURANG(12) = 0
KURANG(13) = 1
KURANG(14) = 1
KURANG(15) = 1
KURANG(16) = 5
X= 0
SIGMA KURANG(i) + X = 16
=====
15-puzzle dapat diselesaikan!
MULAI PENCARIAN...
PENCARIAN SELESAI
=====
Posisi awal:
1 3 4 8
5 2 7 6
9 10 - 11
13 14 15 12
=====
Aksi ke-1: down
Hasil setelah aksi:
1 3 4 8
5 2 - 6
9 10 7 11
13 14 15 12
=====
Aksi ke-2: left
Hasil setelah aksi:
1 3 4 8
5 2 6 -
9 10 7 11
13 14 15 12
=====

```

```

Aksi ke-3: down
Hasil setelah aksi:
1 3 4 -
5 2 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-4: right
Hasil setelah aksi:
1 3 - 4
5 2 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-5: right
Hasil setelah aksi:
1 - 3 4
5 2 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-6: up
Hasil setelah aksi:
1 2 3 4
5 - 6 8
9 10 7 11
13 14 15 12
=====
Aksi ke-7: left
Hasil setelah aksi:
1 2 3 4
5 6 - 8
9 10 7 11
13 14 15 12
=====
Aksi ke-8: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 - 11
13 14 15 12
=====
Aksi ke-9: left
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 11 -
13 14 15 12
=====
Aksi ke-10: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 -
=====
Langkah yang dibutuhkan: 10
Waktu eksekusi program: 0.0
Jumlah simpul yang dibangkitkan: 75

```

5. solved_3.in. Membutuhkan 30 langkah.

```
PS D:\ITB\Sem. 4\Stima\Tucil\Tucil3_13520009> python 15-puzzle-solver.py
Pilih opsi pembangkitan posisi awal 15-Puzzle
1. Dari file
2. Dari random generator
Masukkan nomor opsi: 1
Masukkan path file input: test/solved_3.in
=====
```

input

```

Matriks posisi awal 15-puzzle:
1 3 4 8
9 14 11 -
13 7 5 12
6 2 15 10
Berikut nilai KURANG(i) dan X:
KURANG(1) = 0
KURANG(2) = 0
KURANG(3) = 1
KURANG(4) = 1
KURANG(5) = 1
KURANG(6) = 1
KURANG(7) = 3
KURANG(8) = 4
KURANG(9) = 4
KURANG(10) = 0
KURANG(11) = 5
KURANG(12) = 3
KURANG(13) = 6
KURANG(14) = 8
KURANG(15) = 1
KURANG(16) = 8
X= 0
SIGMA KURANG(i) + X = 46
=====
15-puzzle dapat diselesaikan!
MULAI Pencarian...
PENCARIAN SELESAI
=====
Posisi awal:
1 3 4 8
9 14 11 -
13 7 5 12
6 2 15 10
=====
Aksi ke-1: right
Hasil setelah aksi:
1 3 4 8
9 14 - 11
13 7 5 12
6 2 15 10
=====
Aksi ke-2: right
Hasil setelah aksi:
1 3 4 8
9 - 14 11
13 7 5 12
6 2 15 10
=====
Aksi ke-3: up
Hasil setelah aksi:
1 3 4 8
9 7 14 11
13 - 5 12
6 2 15 10
=====
Aksi ke-4: left
Hasil setelah aksi:
1 3 4 8
9 7 14 11
13 5 - 12
6 2 15 10
=====
Aksi ke-5: down
Hasil setelah aksi:
1 3 4 8
9 7 - 11
13 5 14 12
6 2 15 10
=====

```

```

Aksi ke-6: right
Hasil setelah aksi:
1 3 4 8
9 - 7 11
13 5 14 12
6 2 15 10
=====
Aksi ke-7: up
Hasil setelah aksi:
1 3 4 8
9 5 7 11
13 - 14 12
6 2 15 10
=====
Aksi ke-8: up
Hasil setelah aksi:
1 3 4 8
9 5 7 11
13 2 14 12
6 - 15 10
=====
Aksi ke-9: right
Hasil setelah aksi:
1 3 4 8
9 5 7 11
13 2 14 12
- 6 15 10
=====
Aksi ke-10: down
Hasil setelah aksi:
1 3 4 8
9 5 7 11
- 2 14 12
13 6 15 10
=====
Aksi ke-11: down
Hasil setelah aksi:
1 3 4 8
- 5 7 11
9 2 14 12
13 6 15 10
=====
Aksi ke-12: left
Hasil setelah aksi:
1 3 4 8
5 - 7 11
9 2 14 12
13 6 15 10
=====
Aksi ke-13: up
Hasil setelah aksi:
1 3 4 8
5 2 7 11
9 - 14 12
13 6 15 10
=====
Aksi ke-14: up
Hasil setelah aksi:
1 3 4 8
5 2 7 11
9 6 14 12
13 - 15 10
=====
Aksi ke-15: left
Hasil setelah aksi:
1 3 4 8
5 2 7 11
9 6 14 12
13 15 - 10
=====

```

```

Aksi ke-16: left
Hasil setelah aksi:
1 3 4 8
5 2 7 11
9 6 14 12
13 15 10 -
=====
Aksi ke-17: down
Hasil setelah aksi:
1 3 4 8
5 2 7 11
9 6 14 -
13 15 10 12
=====
Aksi ke-18: down
Hasil setelah aksi:
1 3 4 8
5 2 7 -
9 6 14 11
13 15 10 12
=====
Aksi ke-19: down
Hasil setelah aksi:
1 3 4 -
5 2 7 8
9 6 14 11
13 15 10 12
=====
Aksi ke-20: right
Hasil setelah aksi:
1 3 - 4
5 2 7 8
9 6 14 11
13 15 10 12
=====
Aksi ke-21: right
Hasil setelah aksi:
1 - 3 4
5 2 7 8
9 6 14 11
13 15 10 12
=====
Aksi ke-22: up
Hasil setelah aksi:
1 2 3 4
5 - 7 8
9 6 14 11
13 15 10 12
=====
Aksi ke-23: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 - 14 11
13 15 10 12
=====
Aksi ke-24: left
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 14 - 11
13 15 10 12
=====
Aksi ke-25: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 14 10 11
13 15 - 12

```

```

Aksi ke-25: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 14 10 11
13 15 - 12
=====

```

```

Aksi ke-26: right
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 14 10 11
13 - 15 12
=====

```

```

Aksi ke-27: down
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 - 10 11
13 14 15 12
=====

```

```

Aksi ke-28: left
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 - 11
13 14 15 12
=====

```

```

Aksi ke-29: left
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 11 -
13 14 15 12
=====

```

```

Aksi ke-30: up
Hasil setelah aksi:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 -
=====

```

```

Langkah yang dibutuhkan: 30
Waktu eksekusi program: 34.460206270217896
Jumlah simpul yang dibangkitkan: 1989963

```


output

D. Alamat Github

https://github.com/ARomygithub/Tucil3_13520009

E. Tabel Ceklist

Poin	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	✓	
5. Bonus dibuat		✓