

Algoritma Branch and Bound untuk Menyelesaikan Puzzle 15

Laporan Tugas Kecil



Oleh:

Muhammad Risqi Firdaus 13520043

**TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2022**

A. Algoritma *Brute Force*

Branch and Bound merupakan algoritma turunan dari algoritma pencarian dasar BFS dan DFS. Algoritma ini mengurutkan pencariannya berdasarkan cost yang ada pada tiap node. Sistem yang digunakan adalah priority queue. Elemen dengan nilai cost terendah akan didahulukan.

Cost sendiri merupakan penjumlahan dari nilai total dari fungsi kurang(i) pada puzzle ditambah kedalaman node. Fungsi kurang(i) merupakan fungsi untuk menghitung ada berapa banyak elemen yang lebih kecil dari elemen ke (bar,kol) yang berada pada posisi (x,y) di mana $x > \text{bar}$ atau ($y > \text{kol}$ dan $x = \text{bar}$). Kedalaman node, dimulai dari 0 untuk parent atau root. Selanjutnya, bertambah 1 untuk tiap tingkatan.

Untuk mengoptimalkan program, maka digunakanlah metode heuristik dan prevensi non-solvable. Prevensi non-solvable ditentukan jika nilai kurang i pada puzzle akar ditambah nilai koordinat x dan y bernilai ganjil. Artinya, hanya puzzle dengan nilai kurang i ditambah nilai koordinat x ditambah koordinat y yang genap saja yang dapat diselesaikan.

Ada beberapa metode heuristik yang dapat diterapkan. Pertama, simpan susunan puzzle yang telah dijelajahi dalam dictionary. Dengan begitu, program tidak akan mengecek lagi susunan puzzle yang telah dicek sebelumnya. Kemudian, nilai cost cukup dihitung setelah matrix digerakkan. Dengan begitu diharapkan program dapat dijalankan dengan lebih efisien, tetapi tepat mencapai target untuk menyelesaikan puzzle.

Dalam perancangannya, program terdiri dari dua modul utama, yakni Node berupa object yang menyimpan puzzle dalam matrix, list of previous move, serta cost, dan modul priority queue yang menyimpan queue of puzzleurut berdasarkan cost terkecilnya. Kedua modul dipanggil ke program utama yakni main.py.

Program dimulai dengan memasukkan nama file yang ingin dibaca, selanjutnya program akan menyelesaikan puzzle masukan. Lama waktu penyelesaian tergantung pada kompleksitas puzzle yang diselesaikan, semakin sedikit langkah yang dieksplorasi semakin cepat program dieksekusi.

B. Source Program

Program ditulis dengan bahasa python dengan bantuan modul numpy dan time. Program terdiri atas 3 modul yakni main.py, Node.py dan PrioQueue.py.

Main.py:

```
import numpy as np
from Node import Node
from PrioQueue import PrioQueue
import time

finalMat = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]])

print("+=====+")
print("|| Welcome to 15th Puzzle Solver||")
print("+=====+")
```

```

print()

final = Node(0, [], finalMat)
print("=====")
fileName = input("\tMasukkan Nama File Data Puzzle: ")
print("=====")
f = open(fileName, "r")
arr = []
txt= f.readlines()
for i in txt:
    arr+=[i.split(" ")]
arr=[[int(j.replace('\n','')) for j in i]for i in arr]
f.close()

npArr = np.array(arr)

print()
print("Puzzle Awal:\n",npArr)
print()
puzzleAwal = Node(0, [], npArr)
print("Nilai Kurang (i) dari Puzzle adalah: ")
puzzleAwal.printKurangI()
print()
x,y= np.where(npArr == 16)
print("Nilai \Sigma kurang (i) + X =
{}".format(puzzleAwal.countCost()+((x[0]+y[0])%2)))

startTime = time.time()
if((puzzleAwal.countCost()+x[0]+y[0])%2!=0):
    print("Persoalan tidak dapat diselesaikan :\"")
else:
    prio = PrioQueue()
    prio.append(puzzleAwal)
    arr={}
    arr[np.array_str(puzzleAwal.getMat())]=True
    while(not final.isEqualWith(prio.getHead()) and prio.getLen()>0):
        temp = prio.pop()
        y,x=np.where(temp.getMat()==16)
        if(y[0]>0):

```

```

        node3 =
Node(temp.getDepth()+1,temp.getPrevMove(),temp.getMat())
        node3.move("up")
        if(not (np.array_str(node3.getMat()) in arr)):
            prio.append(node3)
            arr[np.array_str(node3.getMat())]=True
    if (x[0]>0):
        node2 =
Node(temp.getDepth()+1,temp.getPrevMove(),temp.getMat())
        node2.move("left")
        if(not (np.array_str(node2.getMat()) in arr)):
            prio.append(node2)
            arr[np.array_str(node2.getMat())]=True
    if(y[0]<3):
        node1 =
Node(temp.getDepth()+1,temp.getPrevMove(),temp.getMat())
        node1.move("down")
        if(not (np.array_str(node1.getMat()) in arr)):
            prio.append(node1)
            arr[np.array_str(node1.getMat())]=True
    if(x[0]<3):
        node4 =
Node(temp.getDepth()+1,temp.getPrevMove(),temp.getMat())
        node4.move("right")
        if(not (np.array_str(node4.getMat()) in arr)):
            prio.append(node4)
            arr[np.array_str(node4.getMat())]=True

    end = time.time()
    count=0
    for i in (prio.getHead().getPrevMove()):
        print("\nStep {} -> {}".format(count+1,i))
        puzzleAwal.move(i)
        print(puzzleAwal.getMat())
        count+=1

    print("\nPuzzle solved in {} steps\n{} node discovered in {:.5f}
seconds\n".format(len(prio.getHead().getPrevMove()),len(arr),end-startTime
))

```

Node.py:

[illegible]

```

        count+=1
    else:
        for n in range (j,4):
            if(self.mat[i][j]>self.mat[m][n]):
                count+=1
    return count+self.depth

def printKurangI(self):
    for i in range (4):
        for j in range(4):
            count=0
            for m in range (i,4):
                if(m!=i):
                    for n in range (4):
                        if(self.mat[i][j]>self.mat[m][n]):
                            count+=1
                else:
                    for n in range (j,4):
                        if(self.mat[i][j]>self.mat[m][n]):
                            count+=1
            print("Nilai kurang dari ({})) adalah
{}").format(self.mat[i][j],count))

def getDepth(self):
    return self.depth

def getCost(self):
    return self.cost

def getMat(self):
    return self.mat

def getPrevMove(self):
    return self.prevMove

def isEqualWith(self,bro):
    if(bro==None):
        return False
    else:
        return np.array_equal(self.mat,bro.mat)

```

PrioQueue.py:

```
from Node import Node

class PrioQueue:
    def __init__(self) :
        self.queue=[]

    def append(self,puzz):
        #append an elemnt to priotiy queue
        #based on cost
        #if cost is less than the first element in the queue
        #then append to the first element
        #else append to the next element
        #if the queue is empty then append to the first element
        if(len(self.queue)>0):
            i =0
            iter = self.queue[i]
            while(puzz.getCost()>iter.getCost() and i<len(self.queue)-1):
                i+=1
                iter = self.queue[i]
            if(self.queue[i].getCost()<puzz.getCost()):
                self.queue.append(puzz)
            else:
                self.queue.insert(i,puzz)
        else:
            self.queue+= [puzz]

    def prinPrio(self):
        for i in self.queue:
            print(i.getMat())
            print(i.getCost())

    def pop(self):
        return self.queue.pop(0)

    def getHead(self):
        if(len(self.queue)>0):
            return self.queue[0]
        else:
            return None

    def getLen(self):
        return len(self.queue)
```

C. Source Program

1. Sukses

No	Puzzle	Jawaban
1	2 3 4 7 5 6 16 8 9 10 1 15 13 14 12 11	<div> <p> Nilai Kurang (i) dari Puzzle adalah: Nilai kurang dari (2) adalah 1 Nilai kurang dari (3) adalah 1 Nilai kurang dari (4) adalah 1 Nilai kurang dari (7) adalah 3 Nilai kurang dari (5) adalah 1 Nilai kurang dari (6) adalah 1 Nilai kurang dari (16) adalah 9 Nilai kurang dari (8) adalah 1 Nilai kurang dari (9) adalah 1 Nilai kurang dari (10) adalah 1 Nilai kurang dari (1) adalah 0 Nilai kurang dari (15) adalah 4 Nilai kurang dari (13) adalah 2 Nilai kurang dari (14) adalah 2 Nilai kurang dari (12) adalah 1 Nilai kurang dari (11) adalah 0 Nilai \Sigma kurang (i) + X = 30 </p> </div> <div> <p> Step 1 -> down: [[2 3 4 7] [5 6 1 8] [9 10 16 15] [13 14 12 11]] </p> <p> Step 2 -> down: [[2 3 4 7] [5 6 1 8] [9 10 12 15] [13 14 16 11]] </p> <p> Step 3 -> right: [[2 3 4 7] [5 6 1 8] [9 10 12 15] [13 14 11 16]] </p> <p> Step 4 -> up: [[2 3 4 7] [5 6 1 8] [9 10 12 16] [13 14 11 15]] </p> <p> Step 5 -> up: [[2 3 4 7] [5 6 1 16] [9 10 12 8] [13 14 11 15]] </p> </div> <div> <p> Step 6 -> up: [[2 3 4 16] [5 6 1 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 7 -> left: [[2 3 16 4] [5 6 1 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 8 -> left: [[2 16 3 4] [5 6 1 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 9 -> left: [[16 2 3 4] [5 6 1 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 10 -> down: [[5 2 3 4] [16 6 1 7] [9 10 12 8] [13 14 11 15]] </p> </div> <div> <p> Step 11 -> right: [[5 2 3 4] [6 16 1 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 12 -> right: [[5 2 3 4] [6 1 16 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 13 -> up: [[5 2 16 4] [6 1 3 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 14 -> left: [[5 16 2 4] [6 1 3 7] [9 10 12 8] [13 14 11 15]] </p> <p> Step 15 -> down: [[5 1 2 4] [6 16 3 7] [9 10 12 8] [13 14 11 15]] </p> </div>


```
Step 16 -> left:
[[ 5  1  2  4]
 [16  6  3  7]
 [ 9 10 12  8]
 [13 14 11 15]]
```

```
Step 17 -> up:
[[16  1  2  4]
 [ 5  6  3  7]
 [ 9 10 12  8]
 [13 14 11 15]]
```

```
Step 18 -> right
[[ 1 16  2  4]
 [ 5  6  3  7]
 [ 9 10 12  8]
 [13 14 11 15]]
```

```
Step 19 -> right
[[ 1  2 16  4]
 [ 5  6  3  7]
 [ 9 10 12  8]
 [13 14 11 15]]
```

```
Step 20 -> down:
[[ 1  2  3  4]
 [ 5  6 16  7]
 [ 9 10 12  8]
 [13 14 11 15]]
```

```
Step 21 -> right:
[[ 1  2  3  4]
 [ 5  6  7 16]
 [ 9 10 12  8]
 [13 14 11 15]]
```

```
Step 22 -> down:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 12 16]
 [13 14 11 15]]
```

```
Step 23 -> left:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 16 12]
 [13 14 11 15]]
```

```
Step 24 -> down:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 16 15]]
```

```
Step 25 -> right:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Puzzle solved in 25 steps
154211 node discovered in 705.13859 seconds

2	<pre> 1 2 3 4 5 6 7 8 10 11 12 16 9 13 14 15 </pre>	<pre> Puzzle Awal: [[1 2 3 4] [5 6 7 8] [10 11 12 16] [9 13 14 15]] Nilai Kurang (i) dari Puzzle adalah: Nilai kurang dari (1) adalah 0 Nilai kurang dari (2) adalah 0 Nilai kurang dari (3) adalah 0 Nilai kurang dari (4) adalah 0 Nilai kurang dari (5) adalah 0 Nilai kurang dari (6) adalah 0 Nilai kurang dari (7) adalah 0 Nilai kurang dari (8) adalah 0 Nilai kurang dari (10) adalah 1 Nilai kurang dari (11) adalah 1 Nilai kurang dari (12) adalah 1 Nilai kurang dari (16) adalah 4 Nilai kurang dari (9) adalah 0 Nilai kurang dari (13) adalah 0 Nilai kurang dari (14) adalah 0 Nilai kurang dari (15) adalah 0 Nilai \Sigma kurang (i) + X = 8 Step 1 -> left: [[1 2 3 4] [5 6 7 8] [10 11 16 12] [9 13 14 15]] Step 2 -> left: [[1 2 3 4] [5 6 7 8] [10 16 11 12] [9 13 14 15]] Step 3 -> left: [[1 2 3 4] [5 6 7 8] [16 10 11 12] [9 13 14 15]] Step 4 -> down: [[1 2 3 4] [5 6 7 8] [9 10 11 12] [16 13 14 15]] </pre>
---	---	--

		<pre> Step 5 -> right: [[1 2 3 4] [5 6 7 8] [9 10 11 12] [13 16 14 15]] Step 6 -> right: [[1 2 3 4] [5 6 7 8] [9 10 11 12] [13 14 16 15]] Step 7 -> right: [[1 2 3 4] [5 6 7 8] [9 10 11 12] [13 14 15 16]] Puzzle solved in 7 steps 34 node discovered in 0.03979 seconds </pre>
3	<pre> 5 1 2 4 9 6 3 8 16 10 7 11 13 14 15 12 </pre>	<pre> Puzzle Awal: [[5 1 2 4] [9 6 3 8] [16 10 7 11] [13 14 15 12]] Nilai Kurang (i) dari Puzzle adalah: Nilai kurang dari (5) adalah 4 Nilai kurang dari (1) adalah 0 Nilai kurang dari (2) adalah 0 Nilai kurang dari (4) adalah 1 Nilai kurang dari (9) adalah 4 Nilai kurang dari (6) adalah 1 Nilai kurang dari (3) adalah 0 Nilai kurang dari (8) adalah 1 Nilai kurang dari (16) adalah 7 Nilai kurang dari (10) adalah 1 Nilai kurang dari (7) adalah 0 Nilai kurang dari (11) adalah 0 Nilai kurang dari (13) adalah 1 Nilai kurang dari (14) adalah 1 Nilai kurang dari (15) adalah 1 Nilai kurang dari (12) adalah 0 Nilai \Sigma kurang (i) + X = 22 </pre>

```
Step 1 -> up:
[[ 5  1  2  4]
 [16  6  3  8]
 [ 9 10  7 11]
 [13 14 15 12]]
```

```
Step 2 -> up:
[[16  1  2  4]
 [ 5  6  3  8]
 [ 9 10  7 11]
 [13 14 15 12]]
```

```
Step 3 -> right:
[[ 1 16  2  4]
 [ 5  6  3  8]
 [ 9 10  7 11]
 [13 14 15 12]]
```

```
Step 4 -> right:
[[ 1  2 16  4]
 [ 5  6  3  8]
 [ 9 10  7 11]
 [13 14 15 12]]
```

```
Step 5 -> down:
[[ 1  2  3  4]
 [ 5  6 16  8]
 [ 9 10  7 11]
 [13 14 15 12]]
```

```
Step 6 -> down:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 16 11]
 [13 14 15 12]]
```

```
Step 7 -> right:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 16]
 [13 14 15 12]]
```

```
Step 8 -> down:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

```
Puzzle solved in 8 steps
212 node discovered in 0.10807 seconds
```

2. Gagal

No	Puzzle	Jawaban
1	16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	<pre>Puzzle Awal: [[16 1 2 3] [4 5 6 7] [8 9 10 11] [12 13 14 15]] Nilai Kurang (i) dari Puzzle adalah: Nilai kurang dari (16) adalah 15 Nilai kurang dari (1) adalah 0 Nilai kurang dari (2) adalah 0 Nilai kurang dari (3) adalah 0 Nilai kurang dari (4) adalah 0 Nilai kurang dari (5) adalah 0 Nilai kurang dari (6) adalah 0 Nilai kurang dari (7) adalah 0 Nilai kurang dari (8) adalah 0 Nilai kurang dari (9) adalah 0 Nilai kurang dari (10) adalah 0 Nilai kurang dari (11) adalah 0 Nilai kurang dari (12) adalah 0 Nilai kurang dari (13) adalah 0 Nilai kurang dari (14) adalah 0 Nilai kurang dari (15) adalah 0 Nilai \Sigma kurang (i) + X = 15 Persoalan tidak dapat diselesaikan :"(</pre>

2	11 10 8 5 3 9 1 13 14 12 15 7 2 6 4 16	<pre> Puzzle Awal: [[11 10 8 5] [3 9 1 13] [14 12 15 7] [2 6 4 16]] Nilai Kurang (i) dari Puzzle adalah: Nilai kurang dari (11) adalah 10 Nilai kurang dari (10) adalah 9 Nilai kurang dari (8) adalah 7 Nilai kurang dari (5) adalah 4 Nilai kurang dari (3) adalah 2 Nilai kurang dari (9) adalah 5 Nilai kurang dari (1) adalah 0 Nilai kurang dari (13) adalah 5 Nilai kurang dari (14) adalah 5 Nilai kurang dari (12) adalah 4 Nilai kurang dari (15) adalah 4 Nilai kurang dari (7) adalah 3 Nilai kurang dari (2) adalah 0 Nilai kurang dari (6) adalah 1 Nilai kurang dari (4) adalah 0 Nilai kurang dari (16) adalah 0 Nilai \Sigma kurang (i) + X = 59 Persoalan tidak dapat diselesaikan :"(</pre>
---	---	--

Link:

https://github.com/mrfirdauss-20/Tucil-Stima2022/tree/main/Tucil3_13520043

Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi	V	
2. Program Berhasil <i>running</i>	V	
3. Program dapat menerima input dan menuliskan output.	V	
4. Luaran sudah benar untuk semua data uji	V	
5. Bonus		V