

LAPORAN TUGAS KECIL I



Nama: Muhammad Atpur Rafif

NIM: 13522086

Kelas: K2

INSTITUT TEKNOLOGI BANDUNG

2023

Algoritma

Langkah program:

- a. Program memiliki sebuah *state* yang berisi komponen sebagai berikut:
 - i. *board*: komponen permainan seperti matriks dan ukurannya
 - ii. *sequence*: *list* token yang sudah diambil, (nilai awal: *list* kosong)
 - iii. *isHorizontal*: menentukan arah pengambilan token, (nilai awal: *true*)
 - iv. *rcPos*: menentukan posisi pengambilan token berdasarkan arah, (nilai awal: 0)
 - v. *taken*: menandai apabila sebuah koordinat pada matriks telah digunakan
 - vi. *steps*: *list* koordinat pada matriks yang telah diambil (nilai awal: *list* kosong)
- b. Terdapat sebuah variabel yang berisi langkah paling optimal.
- c. Pada pemanggilan pertama fungsi *solve*, dipilih kemungkinan pertama koordinat dan token. Kedua nilai tersebut dimasukkan pada *state* *sequence* dan *steps*.
- d. Perhitungan nilai hadiah dilakukan, apabila nilai tersebut lebih besar daripada nilai optimal saat ini, maka nilai optimal diubah dengan *state* saat ini.
- e. Nilai dari *state* *isHorizontal*, *rcPos* disimpan pada sebuah variabel baru. Kemudian diperbarui dengan mengubah nilai *isHorizontal* dengan kebalikannya, dan *rcPos* dengan nilai berdasarkan koordinat yang dipilih.
- f. Pemanggilan rekursif fungsi *solve* dilakukan hingga panjang dari *state* *sequence* sama dengan panjang *buffer*. Pada saat ini, dilakukan perhitungan nilai hadiah seperti pada langkah d.
- g. Nilai *state* dikembalikan dengan nilai sebelumnya yang telah disimpan pada langkah e.
- h. Fungsi yang dipanggil telah selesai, maka dilanjutkan dengan instruksi pada fungsi yang memanggil. Pada tahap ini, apabila masih terdapat kemungkinan koordinat yang masih bisa dipilih berdasarkan *state* *isHorizontal* dan *rcPos*, maka dilakukan pemanggilan fungsi *solve* dan dilanjutkan seperti pada langkah d sampai g, dengan kata lain, dilakukan *backtracking*.
- i. Setelah fungsi *solve* yang dipanggil pertama kali selesai, maka seluruh kemungkinan langkah yang ada telah dicoba, dengan nilai paling optimal disimpan pada sebuah variabel yang berisi langkah paling optimal.

Konsiderasi:

- a. Struktur traversal program adalah DFS (*Depth First Search*). Dibandingkan dengan BFS (*Breadth First Search*), traversal menggunakan DFS tidak memerlukan *state cloning*. Sedangkan BFS memerlukan *cloning* untuk elemen pada *queue*. Hal ini meningkatkan lokalitas memori.
- b. Terdapat implementasi *multithreading* pada program, pada dasarnya sama dengan *singlethreading*, namun ketika panjang *sequence* mencapai sebuah titik yang dinamakan dengan variabel *divergePoint*, pemanggilan rekursif selanjutnya didelegasikan menuju [worker](#). Manajemen *worker* dilakukan oleh *WorkerManager*, dengan sistem penjadwalan *preemptive*, sehingga *main thread* harus menunggu setidaknya satu *worker* yang menganggur untuk melanjutkan pemanggilan fungsi dengan *asynchronous programming*.
- c. Bahasa pemrograman yang dipilih adalah *typescript*, yang nantinya dikompilasi menjadi *javascript*, dan dijalankan oleh Node.js setidaknya versi 21

Program

Potongan kode dalam bahasa *typescript* dibawah hanya mendemonstrasikan algoritma yang telah dijelaskan diatas. Program ini tidak langsung dapat dijalankan, seperti variabel `msg` yang digunakan oleh `state` belum didefinisikan. Perhitungan nilai hadiah dilakukan dengan iterasi biasa. Kode secara keseluruhan dapat dilihat pada repositori.

```
const state = JSON.parse(msg) /* State dari main thread, bisa state awal */
const optimal: FinishedMessage = {
  reward: 0,
  sequence: [],
  steps: [],
  time: 0
}
const runner = () => {
  refreshOptimal(optimal, state)
  if (state.sequence.length == state.board.buffer) return;
  const oldIsHorizontal = state.isHorizontal
  const oldRcPos = state.rcPos
  state.isHorizontal = !state.isHorizontal
  const maxRcPos = oldIsHorizontal ? state.board.width : state.board.height
  for (let newRcPos = 0; newRcPos < maxRcPos; ++newRcPos) {
    const row = oldIsHorizontal ? oldRcPos : newRcPos
    const col = oldIsHorizontal ? newRcPos : oldRcPos
    const flatten = row * state.board.width + col
    if (state.taken[flatten] == true) continue
    const token = state.board.matrix[row][col]
    state.rcPos = newRcPos
    state.sequence.push(token)
    state.steps.push([row, col])
    state.taken[flatten] = true
    runner()
    state.taken[flatten] = false
    state.sequence.pop()
    state.steps.pop()
  }

  state.isHorizontal = oldIsHorizontal
  state.rcPos = oldRcPos
}
const start = performance.now()
runner()
const end = performance.now()
optimal.time = end - start
/* Hasil disimpan pada variabel optimal */
```

Hasil

Tes 1

```

> node ./bin/generator.js
Value inside square bracket mean default, fill empty (just enter) to use that value

Unique token count [5]:
Unique token (separated by space) [BD 1C 7A 55 E9]:
Buffer size [7]:
Board width [6]:
Board height [6]:
Sequence count [3]:
Max sequence length [4]:

7
6 6
1C 7A 7A 1C BD 1C
7A BD BD BD 1C E9
1C BD 1C BD BD E9
55 7A BD 55 7A 1C
55 55 1C BD BD BD
BD E9 55 55 7A 55
3
BD E9 E9
-13
BD 55
-45
E9 7A
-86

Save to file (empty to ignore): ./test/test_1.txt
~/tugas/IF2211-Tucil-1 >

> node ./bin/quickhack.js
File path: ./test/test_1.txt
=== Raw Result ===
{
  reward: 0,
  sequence: [],
  steps: [],
  totalTime: 198.66141599999992,
  totalWorkerTime: 131.153110999999417,
  averageWorkerTime: 0.8743540733332945
}

=== Formatted Result ===
0

198ms

Save to file? (blank to ignore): ./test/result_1.txt
~/tugas/IF2211-Tucil-1 >

```

Tes 2

```

> node ./bin/generator.js
Value inside square bracket mean default, fill empty (just enter) to use that value

Unique token count [5]:
Unique token (separated by space) [BD 1C 7A 55 E9]:
Buffer size [7]:
Board width [6]:
Board height [6]:
Sequence count [3]:
Max sequence length [4]:

7
6 6
55 55 BD 55 1C 7A
7A 7A 7A 1C BD 1C
7A BD 1C 55 55 E9
E9 7A BD 7A 55 7A
55 E9 E9 BD 7A 1C
E9 7A BD 55 1C 7A
3
BD E9
-1
1C E9
54
1C 55
98

Save to file (empty to ignore): ./test/test_2.txt
~/tugas/IF2211-Tucil-1 >

> node ./bin/quickhack.js
File path: ./test/test_2.txt
=== Raw Result ===
{
  reward: 152,
  sequence: [ '1C', '55', '1C', 'E9' ],
  steps: [ [ 0, 4 ], [ 2, 4 ], [ 2, 2 ], [ 4, 2 ] ],
  totalTime: 141.19637999999994,
  totalWorkerTime: 133.384704999999494,
  averageWorkerTime: 0.8092313666666329
}

=== Formatted Result ===
152
1C 55 1C E9
5, 1
5, 3
3, 3
3, 5

141ms

Save to file? (blank to ignore): ./test/result_2.txt
~/tugas/IF2211-Tucil-1 >

```

Tes 3

```
node > node ./bin/generator.js
Value inside square bracket mean default, fill empty (just enter) to use that value

Unique token count [5]:
Unique token (separated by space) [BD 1C 7A 55 E9]:
Buffer size [7]:
Board width [6]:
Board height [6]:
Sequence count [3]:
Max sequence length [4]:

7
6 6
1C BD 7A 7A 7A E9
BD BD 7A 55 7A 55
BD BD 1C 7A 7A 1C
7A BD BD 1C E9 55
55 BD 55 1C BD 1C
BD 7A 7A E9 BD 55
3
E9 E9
99
1C 1C
-51
7A 55
-31

Save to file (empty to ignore): ./test/test_3.txt
~/tugas/IF2211-Tucil-1 >

node > node ./bin/quickhack.js
File path: ./test/test_3.txt
=== Raw Result ===
{
  reward: 0,
  sequence: [],
  steps: [],
  totalTime: 152.06733999999997,
  totalWorkerTime: 171.93558600000526,
  averageWorkerTime: 1.1462367866667019
}

=== Formatted Result ===
0

152ms

Save to file? (blank to ignore): ./test/result_3.txt
~/tugas/IF2211-Tucil-1 >
```

Tes 4

```
node > node ./bin/generator.js
Value inside square bracket mean default, fill empty (just enter) to use that value

Unique token count [5]:
Unique token (separated by space) [BD 1C 7A 55 E9]:
Buffer size [7]: 9
Board width [6]: 9
Board height [6]: 9
Sequence count [3]: 5
Max sequence length [4]: 4

9
9 9
E9 E9 1C E9 7A E9 7A E9 1C
7A BD 55 55 7A 55 55 7A 1C
E9 55 1C BD 7A E9 E9 55 7A
55 55 1C E9 55 E9 BD BD E9
1C BD 7A BD E9 55 E9 BD E9
E9 55 7A 55 BD 1C 55 7A 7A
E9 7A E9 1C BD 55 55 E9 7A
BD 55 1C 7A 55 BD BD BD 55
BD 1C 55 7A 1C 55 7A E9 55
5
55 55 E9
-89
E9 E9
-57
1C 1C E9
-22
1C 55
71
E9 E9 1C
4

Save to file (empty to ignore): ./test/test_4.txt
~/tugas/IF2211-Tucil-1 >

node > node ./bin/quickhack.js
File path: ./test/test_4.txt
=== Raw Result ===
{
  reward: 71,
  sequence: [ '1C', '55' ],
  steps: [ [ 0, 2 ], [ 1, 2 ] ],
  totalTime: 3670.3366689999993,
  totalWorkerTime: 42142.55937800002,
  averageWorkerTime: 73.1641655860856
}

=== Formatted Result ===
71
1C 55
3, 1
3, 2

3670ms

Save to file? (blank to ignore):
```

Tes 5

```
node > node ./bin/generator.js
Value inside square bracket mean default, fill empty (just enter) to use that value

Unique token count [5]:
Unique token (separated by space) [BD 1C 7A 55 E9]:
Buffer size [7]: 15
Board width [6]: 5
Board height [6]: 5
Sequence count [3]: 10
Max sequence length [4]: 4

15
5 5
BD 1C 1C 55 E9
55 E9 1C 55 BD
1C BD 7A 7A 7A
E9 7A E9 7A 7A
55 BD E9 1C 7A
10
55 55
-53
1C E9 E9
83
7A 7A
-18
55 55
-90
55 7A
-49
BD BD
-59
7A E9
16
55 E9
-92
BD E9
-20
1C BD
-67

Save to file (empty to ignore): ./test/test_5.txt
~/tugas/IF2211-Tucil-1 >

zsh @ IF2211-Tucil-1
```

```
node > node ./bin/quickhack.js
File path: ./test/test_5.txt
=== Raw Result ===
{
  reward: 99,
  sequence: [ '1C', 'E9', 'E9', '1C', '7A', 'E9' ],
  steps: [ [ 0, 2 ], [ 3, 2 ], [ 3, 0 ], [ 2, 0 ], [ 2, 2 ], [ 4, 2 ] ],
  totalTime: 3473.6850029999996,
  totalWorkerTime: 30041.575625,
  averageWorkerTime: 475.5171953125
}

=== Formatted Result ===
99
1C E9 E9 1C 7A E9
3, 1
3, 4
1, 4
1, 3
3, 3
3, 5

3473ms

Save to file? (blank to ignore): ./test/result_5.txt
~/tugas/IF2211-Tucil-1 >
```

Tes 6

```
node > node ./bin/generator.js
Unique token count [5]:
Unique token (separated by space) [BD 1C 7A 55 E9]:
Buffer size [7]: 10
Board width [6]: 10
Board height [6]: 10
Sequence count [3]: 10
Max sequence length [4]: 4

10
10 10
BD E9 BD 7A 1C 55 E9 BD E9 7A
1C E9 7A 7A 55 7A 7A E9 55 55
E9 E9 1C E9 55 7A 55 1C 55 BD
E9 BD 55 E9 BD 55 BD 55 55 7A
BD E9 1C 1C 1C E9 1C E9 1C BD
BD 1C 55 55 BD 55 55 BD 7A 1C
E9 1C BD 1C 1C E9 7A 1C 55 55
1C BD 7A BD 1C BD E9 BD 7A BD
E9 BD 1C 55 E9 1C 1C 7A E9 E9
BD 7A BD 55 E9 1C 55 7A BD 1C
10
BD 1C
-84
55 BD
77
E9 55
-37
7A 1C
-72
55 7A
59
BD 7A
52
BD 55
6
E9 E9
96
7A 1C 7A
-50
55 55
-88

Save to file (empty to ignore): ./test/test_6.txt
~/tugas/IF2211-Tucil-1 >

zsh @ IF2211-Tucil-1
```

```
node > node ./bin/quickhack.js
File path: ./test/test_6.txt
=== Raw Result ===
{
  reward: 290,
  sequence: [
    'E9', 'E9', 'BD',
    '55', '7A', '55',
    'BD', '7A'
  ],
  steps: [
    [ 0, 1 ], [ 2, 1 ],
    [ 2, 9 ], [ 1, 9 ],
    [ 1, 2 ], [ 3, 2 ],
    [ 3, 1 ], [ 9, 1 ]
  ],
  totalTime: 154459.396971,
  totalWorkerTime: 1843518.3961919993,
  averageWorkerTime: 2275.948637274073
}

=== Formatted Result ===
290
E9 E9 BD 55 7A 55 BD 7A
2, 1
2, 3
10, 3
10, 2
3, 2
3, 4
2, 4
2, 10

154459ms

Save to file? (blank to ignore): ./test/result_6.txt
~/tugas/IF2211-Tucil-1 >
```

Link Repository

https://github.com/atpur-rafif/Tucil1_13522086