

LAPORAN TUGAS BESAR STRATEGI ALGORITMA
UNLOCK PIN PROBLEM
IF - 45 - 09



ANGGOTA KELOMPOK 2

MOH HILMAN FARIZ	1301213393
M DZAKIYUDDIN HAIDAR	1301213048
RENATA ALFIANTO RASYID	1301213304

FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2023

DAFTAR ISI

DAFTAR ISI.....	1
ABSTRAK.....	2
BAB I.....	3
PENDAHULUAN.....	3
A. LATAR BELAKANG.....	3
B. RUMUSAN MASALAH.....	4
C. TUJUAN.....	4
BAB II.....	5
DASAR TEORI.....	5
A. PENJELASAN UMUM.....	5
B. METODE YANG TERPILIH.....	5
BAB III.....	6
IMPLEMENTASI.....	6
1. Brute Force.....	6
2. Backtracking.....	7
BAB IV.....	7
ANALISIS.....	7
A. BRUTE FORCE.....	7
B. BACKTRACKING.....	8
C. PERBANDINGAN.....	10
BAB V.....	11
KESIMPULAN.....	11
DAFTAR PUSTAKA.....	12
LAMPIRAN.....	13
LINK KODE.....	15

ABSTRAK

Penyelesaian permasalahan PIN Problem 3 digit menggunakan metode Brute Force dan Backtracking adalah fokus utama dalam penelitian ini. Metode Brute Force digunakan untuk mencoba semua kemungkinan kombinasi digit secara berurutan, sedangkan metode backtracking membantu menghindari langkah-langkah yang tidak valid, meningkatkan efisiensi algoritma. Dalam konteks penyelesaian PIN problem 3 digit, metode Backtracking terbukti mampu menghasilkan solusi optimal dengan mencoba semua kemungkinan yang valid.

Pada kompleksitas waktu dari metode Brute Force ini adalah $O(10^n)$ karena jumlah langkah yang diperlukan untuk mencari PIN yang benar tidak bergantung pada ukuran input (dalam hal ini, digit 3). Best Case jika PIN yang dicari langsung ditemukan pada percobaan pertama, dengan kompleksitas $O(1)$, kompleksitas average case dan worst case $O(n^{10})$ dengan mencoba semua kemungkinan yang ada.

Sedangkan kompleksitas waktu dari metode Backtracking ini adalah $O(10^n)$. Jumlah percobaan yang harus dilakukan adalah 10^3 , karena setiap digit dapat memiliki 10 kemungkinan nilai (0 hingga 9), dan kita harus mencoba semua kemungkinan untuk setiap digit. Best Case jika semua digit PIN yang diperiksa langsung cocok dengan array PIN yang disimpan, maka tidak perlu mencoba kombinasi digit lainnya, dengan kompleksitasnya $O(1)$. kompleksitas average case dinyatakan sebagai $O(10 \cdot (n/2))$ dan worst case kompleksitas waktu akan menjadi $O(10 \cdot n)$

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Penyelesaian PIN Problem, seperti mencari kombinasi PIN seringkali melibatkan pendekatan metode Brute Force dan Backtracking. Metode Brute Force adalah teknik sederhana yang mencoba semua kemungkinan kombinasi secara berurutan untuk mencapai solusi yang diinginkan, sedangkan metode Backtracking menggunakan pendekatan rekursif untuk secara sistematis mencoba semua kemungkinan solusi dengan menghindari langkah-langkah yang tidak valid.

Metode Brute Force maupun backtracking dapat digunakan untuk mencari kombinasi yang benar secara berurutan sampai solusi ditemukan. Metode Brute Force akan secara terus-menerus mencoba semua kombinasi secara berturut-turut, sedangkan metode backtracking akan mencoba setiap digit atau karakter pada setiap posisi dan melanjutkan ke posisi berikutnya jika langkah sebelumnya valid.

Kami akan memberikan contoh implementasi kode menggunakan C++ untuk menjelaskan konsep dan langkah-langkah yang terlibat dalam kedua metode tersebut. Kami juga akan membahas kompleksitas waktu yang terkait dengan masing-masing metode, serta mempertimbangkan kelebihan dan kekurangan dari setiap pendekatan.

B. RUMUSAN MASALAH

Rumusan masalah dari penyelesaian permasalahan PIN problem 3 digit menggunakan metode Brute Force dan backtracking adalah sebagai berikut:

1. Bagaimana mencari solusi PIN problem 3 digit secara sistematis
2. Bagaimana mengimplementasikan metode Brute Force untuk mencoba semua kemungkinan kombinasi digit?
3. Bagaimana menerapkan metode backtracking untuk memecahkan permasalahan PIN Problem?
4. Bagaimana kompleksitas waktu dari permasalahan yang diselesaikan dengan kedua metode tersebut

C. TUJUAN

Tujuan dari penyelesaian permasalahan PIN problem 3 digit menggunakan metode Brute Force dan backtracking adalah sebagai berikut:

1. Mencari solusi secara sistematis

Metode Brute Force dan backtracking memungkinkan kita untuk mencari solusi secara sistematis dengan menguji semua kemungkinan secara berurutan. Ini memastikan bahwa tidak ada solusi yang terlewat atau diabaikan.

2. Mendapatkan solusi optimal

Metode Brute Force dan backtracking dapat menghasilkan solusi optimal, yaitu solusi yang memenuhi semua kriteria dan memberikan hasil terbaik. Dalam PIN problem 3 digit, dengan mencoba semua kemungkinan kombinasi, kita dapat menemukan solusi yang tepat tanpa meninggalkan kemungkinan yang valid.

3. Memahami kompleksitas masalah

Dengan menerapkan metode Brute Force dan backtracking pada permasalahan PIN problem 3 digit, kita dapat memahami kompleksitas masalah dan mempelajari tentang algoritma dan strategi penyelesaiannya. Hal ini dapat membantu kita dalam memecahkan masalah yang lebih kompleks di bidang lain.

BAB II

DASAR TEORI

A. PENJELASAN UMUM

Kasus yang kami pilih ialah pemecahan PIN digital 3 digit atau “Unlock PIN Problem”. Kasus “Unlock PIN Problem” umumnya merupakan kasus ketika seseorang lupa dengan PIN miliknya namun ia membutuhkan PIN tersebut untuk dapat mengakses atau membuka perangkat tertentu. Kasus ini biasanya terkait dengan keamanan perangkat elektronik seperti ponsel, tablet, atau perangkat lain.

B. METODE YANG TERPILIH

1. Brute Force

Metode Brute Force adalah teknik sederhana yang mencoba semua kemungkinan kombinasi secara berurutan untuk mencapai solusi yang diinginkan. Metode ini dipilih karena dapat menyelesaikan kasus ini dengan cara mencoba semua kombinasi secara berturut-turut.

2. Backtracking

Metode Backtracking menggunakan pendekatan rekursif untuk secara sistematis mencoba semua kemungkinan solusi dengan menghindari langkah-langkah yang tidak valid. Metode ini dipilih karena dapat menyelesaikan kasus ini dengan cara mencoba setiap digit atau karakter pada setiap posisi dan dapat melanjutkan ke posisi berikutnya jika langkah sebelumnya valid.

BAB III

IMPLEMENTASI

Penerapan masing-masing metode untuk kasus “Unlock PIN Problem” memiliki perbedaan yang cukup signifikan. Kemudian saat di implementasikan ke dalam kode akan memiliki perbedaan saat diimplementasikan di dunia nyata. Berikut penjelasan dari masing - masing metode.

1. Brute Force

Implementasi pemecahan permasalahan “Unlock PIN Problem” menggunakan metode Brute Force akan dilakukan dengan mencoba semua kemungkinan kombinasi, maka saat mencoba semua kemungkinan kombinasi PIN 3 digit yaitu dari 000 sampai PIN ditemukan akan mengeluarkan output yang sudah ditentukan. Namun, jika PIN yang dicari tidak valid sampai digit 999 maka program yang kami buat tidak akan mengeluarkan output apapun. Meskipun didalam program yang kami buat, kami membuat fungsi generate number untuk menghasilkan nomor acak dari 000 sampai 999. Jadi dalam program yang kami buat, kecil kemungkinan PIN tidak valid, karena PIN sudah ditentukan dari 000 sampai 999.

2. Backtracking

Implementasi pemecahan permasalahan “Unlock PIN Problem” dengan menggunakan metode backtracking adalah dengan mencari PIN dari 0 - 9 pada setiap digit. Kemudian, jika PIN pada digit pertama sudah ditemukan, maka pencarian akan dilanjutkan ke digit berikutnya. Dalam pengimplementasian kodenya kami menggunakan array, dimana setiap digitnya akan diubah menjadi array. Sehingga saat pengecekan akan dibandingkan antara nilai array yang dicari dengan nilai yang sudah ditentukan oleh fungsi generate PIN. Kemudian, saat sudah sesuai maka akan ke array selanjutnya hingga array habis.

BAB IV

ANALISIS

A. BRUTE FORCE

Dalam metode brute force, semua kemungkinan kombinasi 3 digit akan dicoba secara berurutan hingga PIN yang benar ditemukan. Sehingga kompleksitas waktu dari metode brute force ini akan berbeda di setiap casenya, penjelasannya sebagai berikut:

- Best Case:

Dalam skenario terbaik, jika PIN yang dicari langsung ditemukan pada percobaan pertama, kompleksitas waktu akan menjadi $O(1)$, karena hanya satu percobaan yang diperlukan.

- Average Case:

Dalam skenario rata-rata, kita harus mencoba semua kemungkinan kombinasi digit untuk menemukan PIN yang benar. Jumlah percobaan yang perlu dilakukan adalah 10^n , karena setiap digit memiliki 10 kemungkinan nilai (0 hingga 9) dan kita harus mencoba semua kemungkinan kombinasi digit. Oleh karena itu, kompleksitas waktu rata-rata adalah $O(10^n)$.

- Worst Case:

Dalam skenario terburuk, PIN yang dicari adalah yang terakhir diuji setelah mencoba semua kemungkinan kombinasi digit. Dalam kasus ini, kompleksitas waktu tetap $O(10^n)$, karena kita harus mencoba semua kemungkinan untuk setiap digit pada PIN.

Jadi, kompleksitas waktu terbaik (best case) adalah $O(1)$, kompleksitas waktu rata-rata (average case) adalah $O(10^n)$, dan kompleksitas waktu terburuk (worst case) adalah $O(10^n)$

Berikut contoh pencarian secara brute force dengan nilai PIN 125,


```

===== Brute Force =====
current test PIN = 1
current test PIN = 2
current test PIN = 3
current test PIN = 4
current test PIN = 5
current test PIN = 6
current test PIN = 7
current test PIN = 8
current test PIN = 9
current test PIN = 10
current test PIN = 11
current test PIN = 12
current test PIN = 13
current test PIN = 14
current test PIN = 15
current test PIN = 16
current test PIN = 17
current test PIN = 18
current test PIN = 19
current test PIN = 20
current test PIN = 21
current test PIN = 22
current test PIN = 23
current test PIN = 24
current test PIN = 25
current test PIN = 26
current test PIN = 27
current test PIN = 28
current test PIN = 29
current test PIN = 30
current test PIN = 31
current test PIN = 32
current test PIN = 33
current test PIN = 34
current test PIN = 35
current test PIN = 36
current test PIN = 37
current test PIN = 38
current test PIN = 39
current test PIN = 40
current test PIN = 41
current test PIN = 42
current test PIN = 43
current test PIN = 44
current test PIN = 45
current test PIN = 46

```

```

current test PIN = 47
current test PIN = 48
current test PIN = 49
current test PIN = 50
current test PIN = 51
current test PIN = 52
current test PIN = 53
current test PIN = 54
current test PIN = 55
current test PIN = 56
current test PIN = 57
current test PIN = 58
current test PIN = 59
current test PIN = 60
current test PIN = 61
current test PIN = 62
current test PIN = 63
current test PIN = 64
current test PIN = 65
current test PIN = 66
current test PIN = 67
current test PIN = 68
current test PIN = 69
current test PIN = 70
current test PIN = 71
current test PIN = 72
current test PIN = 73
current test PIN = 74
current test PIN = 75
current test PIN = 76
current test PIN = 77
current test PIN = 78
current test PIN = 79
current test PIN = 80
current test PIN = 81
current test PIN = 82
current test PIN = 83
current test PIN = 84
current test PIN = 85
current test PIN = 86
current test PIN = 87
current test PIN = 88
current test PIN = 89
current test PIN = 90
current test PIN = 91
current test PIN = 92
current test PIN = 93
current test PIN = 94

```

```

current test PIN = 95
current test PIN = 96
current test PIN = 97
current test PIN = 98
current test PIN = 99
current test PIN = 100
current test PIN = 101
current test PIN = 102
current test PIN = 103
current test PIN = 104
current test PIN = 105
current test PIN = 106
current test PIN = 107
current test PIN = 108
current test PIN = 109
current test PIN = 110
current test PIN = 111
current test PIN = 112
current test PIN = 113
current test PIN = 114
current test PIN = 115
current test PIN = 116
current test PIN = 117
current test PIN = 118
current test PIN = 119
current test PIN = 120
current test PIN = 121
current test PIN = 122
current test PIN = 123
current test PIN = 124
current test PIN = 125

your PIN : 125 (PIN guessed)
time taken : 71 ms

```

B. BACKTRACKING

Dalam metode backtracking, kombinasi 3 digit akan dicoba semua kemungkinan solusi dengan menghindari langkah-langkah yang tidak valid hingga menemukan solusi yang benar. Sehingga kompleksitas waktu dari metode brute force ini akan berbeda di setiap casenya, penjelasannya sebagai berikut:

- Best Case:

Dalam skenario terbaik, jika semua digit PIN yang diperiksa langsung cocok dengan array PIN yang disimpan, maka tidak perlu mencoba kombinasi digit lainnya. Dalam kasus ini, kompleksitas waktu akan menjadi $O(1)$, karena hanya satu percobaan yang diperlukan untuk membandingkan semua digit dengan array PIN yang disimpan.

- Average Case:

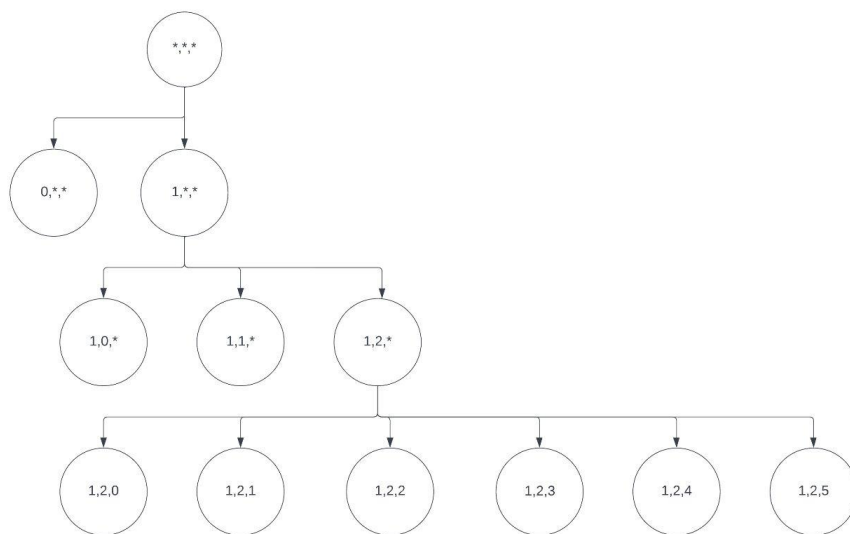
Dalam skenario rata-rata, kita dapat mengasumsikan bahwa digit yang cocok dengan array PIN yang disimpan tersebar secara acak di seluruh posisi digit. Dalam hal ini, rata-rata jumlah percobaan yang perlu dilakukan sekitar setengah dari total kemungkinan kombinasi digit. Oleh karena itu, kompleksitas waktu rata-rata dapat dinyatakan sebagai $O((10 * n)/2)$, di mana n adalah jumlah digit pada PIN.

- Worst Case:

Dalam skenario terburuk, jika tidak ada digit yang cocok dengan array PIN yang disimpan sampai digit terakhir, maka semua kemungkinan kombinasi digit harus diuji. Dalam kasus ini, kompleksitas waktu akan menjadi $O(10 * n)$, karena kita harus mencoba semua kemungkinan untuk setiap digit pada PIN.

Jadi, kompleksitas waktu terbaik (best case) adalah $O(1)$, kompleksitas waktu rata-rata (average case) adalah $O((10*n)/2)$, dan kompleksitas waktu terburuk (worst case) adalah $O(10*n)$

Berikut contoh space tree dari backtracking dengan nilai PIN 125,



```

===== Backtracking =====
Digit ke-0 = 0 (declined)
Digit ke-0 = 1 (accepted)

Digit ke-1 = 0 (declined)
Digit ke-1 = 1 (declined)
Digit ke-1 = 2 (accepted)

Digit ke-2 = 0 (declined)
Digit ke-2 = 1 (declined)
Digit ke-2 = 2 (declined)
Digit ke-2 = 3 (declined)
Digit ke-2 = 4 (declined)
Digit ke-2 = 5 (accepted)

your PIN   : 125 (PIN guessed)
count      : 11
time taken : 12 ms
  
```

C. PERBANDINGAN

Dari hasil percobaan dan perhitungan yang kami lakukan, didapatkan hasil sebagai berikut,

1. Brute Force

Kompleksitas waktu terbaik (best case) adalah $O(1)$, kompleksitas waktu rata-rata (average case) adalah $O(10^n)$, dan kompleksitas waktu terburuk (worst case) adalah $O(10^n)$

2. Backtracking

Kompleksitas waktu terbaik (best case) adalah $O(1)$, kompleksitas waktu rata-rata (average case) adalah $O((10 \cdot n)/2)$, dan kompleksitas waktu terburuk (worst case) adalah $O(10 \cdot n)$

Jadi, dari kedua fakta diatas, metode yang lebih baik untuk kasus “Unlock PIN Problem” adalah metode backtracking karena memiliki tingkat efisiensi yang lebih baik dengan $T(n) \in O(10 \cdot n)$ dibandingkan metode brute force dengan $T(n) \in O(10^n)$.

PIN	BRUTE FORCE	BACKTRACKING
796	401 ms	22 ms
401	216 ms	15 ms
105	97 ms	15 ms

BAB V

KESIMPULAN

Dalam penyelesaian menggunakan metode Brute Force, semua kemungkinan kombinasi digit diuji secara berurutan. Meskipun ini dapat memakan waktu dalam kasus masalah yang lebih kompleks, namun untuk PIN problem 3 digit, waktu yang diperlukan masih dapat diterima. Sedangkan, Dalam penyelesaian menggunakan metode backtracking, langkah-langkah yang tidak valid dibatalkan, sehingga hanya langkah-langkah yang mengarah ke solusi yang valid yang dieksplorasi. Hal ini membantu mengurangi jumlah percobaan yang tidak perlu, meningkatkan efisiensi algoritma.

Kesimpulannya yang kami dapatkan adalah metode backtracking lebih baik digunakan untuk permasalahan “Unlock PIN Problem” karena solusi yang diselesaikan lebih optimal dibandingkan metode Brute Force.

DAFTAR PUSTAKA

- [1] Slide Kuliah Topik 1 : Brute Force
: <https://lms.telkomuniversity.ac.id/mod/resource/view.php?id=2733244>
- [2] Slide Kuliah Topik 5: Backtracking
<https://lms.telkomuniversity.ac.id/mod/url/view.php?id=2733277>
- [3] [SA] Tugas W11_Backtracking_SOLUSI
<https://lms.telkomuniversity.ac.id/mod/resource/view.php?id=3143836>
- [4] <https://www.geeksforgeeks.org/asymptotic-notation-and-analysis-based-on-input-size-of-algorithms/>
- [5] <https://www.geeksforgeeks.org/introduction-to-backtracking-data-structure-and-algorithm-tutorials/>

LAMPIRAN

Backtracking

```
long long backtrack(string tester) {
    auto start_time = high_resolution_clock::now();
    int count = 0;
    string number = "";
    for (int ret = 0; ret < tester.length(); ret++) {
        for (int i = 0; i < 10; i++) {
            cout << "Digit ke-" << ret << " = " << i;
            if (to_string(i) == string(1, tester[ret])) {
                cout << " (accepted)" << endl << endl;
                number += to_string(i);
                count++;
                break;
            }
            else {
                cout << " (declined)" << endl;
                count++;
            }
        }
    }

    cout << "your PIN : " << number << " (PIN guessed)" << endl;
    cout << "count : " << count << endl;
    auto end_time = high_resolution_clock::now();
    auto time_lapsed = duration_cast<milliseconds>(end_time - start_time);
    cout << "time taken : " << time_lapsed.count() << " ms" << endl;
    return time_lapsed.count();
}
```

```

===== Backtracking =====
Digit ke-0 = 0 (declined)
Digit ke-0 = 1 (declined)
Digit ke-0 = 2 (accepted)

Digit ke-1 = 0 (accepted)

Digit ke-2 = 0 (declined)
Digit ke-2 = 1 (declined)
Digit ke-2 = 2 (declined)
Digit ke-2 = 3 (declined)
Digit ke-2 = 4 (declined)
Digit ke-2 = 5 (declined)
Digit ke-2 = 6 (declined)
Digit ke-2 = 7 (accepted)

your PIN    : 207 (PIN guessed)
count       : 12
time taken  : 2 ms

```

Brute Force

```

void bruteforce(int passw) {
    auto start_time = high_resolution_clock::now();
    int guess = 0;
    while (true) {
        guess++;
        cout << "current test PIN = " << guess << endl;
        if (guess == passw) {
            cout << endl;
            cout << "your PIN : " << guess << " (PIN guessed)" << endl;
            break;
        }
    }
    auto end_time = high_resolution_clock::now();
    auto time_lapsed = duration_cast<milliseconds>(end_time - start_time);
    cout << "time taken : " << time_lapsed.count() << " ms" << endl;
}

```

```
current test PIN = 177
current test PIN = 178
current test PIN = 179
current test PIN = 180
current test PIN = 181
current test PIN = 182
current test PIN = 183
current test PIN = 184
current test PIN = 185
current test PIN = 186
current test PIN = 187
current test PIN = 188
current test PIN = 189
current test PIN = 190
current test PIN = 191
current test PIN = 192
current test PIN = 193
current test PIN = 194
current test PIN = 195
current test PIN = 196
current test PIN = 197
current test PIN = 198
current test PIN = 199
current test PIN = 200
current test PIN = 201
current test PIN = 202
current test PIN = 203
current test PIN = 204
current test PIN = 205
current test PIN = 206
current test PIN = 207

your PIN    : 207 (PIN guessed)
time taken : 48 ms
```

LINK KODE:

<https://drive.google.com/file/d/1jF715NddpLhhYav0UcwFvAOjH0qoLIlz/view?usp=sharing>