

Penggunaan Algoritma Brute Force dalam Menyelesaikan Lock Code Puzzle di Ragnarok Online

Felix Setiawan 13518078
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
Felixsetiawan2000@gmail.com

Abstraksi—Lock code puzzle merupakan salah satu jenis persoalan yang menarik untuk diselesaikan. Sampai saat ini belum ada algoritma selain brute force untuk memecahkan persoalan ini. Di game Ragnarok Online, terdapat satu misi yang mengharuskan pemain menebak angka untuk membuka kunci. Pada makalah ini akan dijelaskan penggunaan algoritma brute force untuk menyelesaikan puzzle crack the code pada Ragnarok Online.

Keywords— lock code puzzle; brute force; ragnarok online

I. PENDAHULUAN

Lock code puzzle adalah suatu persoalan umum yang kembali populer setelah belakangan ini banyak dibagikan melalui social media. Pada dasarnya persoalan crack the code merupakan persoalan menebak kata sandi seseorang. Namun persoalan tersebut disederhanakan dengan beberapa cara, antara lain membatasi banyaknya karakter, memberikan petunjuk-petunjuk dan lainnya.

Terdapat beberapa jenis lock code puzzle. Salah satunya merupakan yang banyak dibagikan di social media yaitu, menebak permutasi n angka yang petunjuknya telah diberikan. Jenis kedua adalah menebak permutasi n angka dengan petunjuk yang didapatkan seiring memasukkan angka tersebut.

Jauh sebelum ada media social dan viralnya persoalan ini, terdapat satu game online yang sangat populer di masanya yaitu Ragnarok Online. Sampai saat ini, banyaknya misi yang terdapat di game yang pertama kali dirilis pada tahun 2002 itu sudah tidak terhitung jumlahnya. Di antara banyaknya misi tersebut, salah satu yang menarik perhatian adalah lock code puzzle yang terdapat di Thanatos Tower lantai 4. Lock code puzzle harus diselesaikan agar pemain mendapatkan salah satu kunci untuk memanggil boss monster dengan drop item salah satu termahal di game tersebut hingga saat ini.

Hingga saat ini, belum ditemukan algoritma selain algoritma brute force untuk menyelesaikan lock code puzzle ini. Algoritma brute force cukup tepat digunakan karena domain persoalan yang terbatas dan langkah-langkah yang diambil di setiap tahapnya tidak dapat ditentukan di awal.

II. DASAR TEORI

A. Brute Force

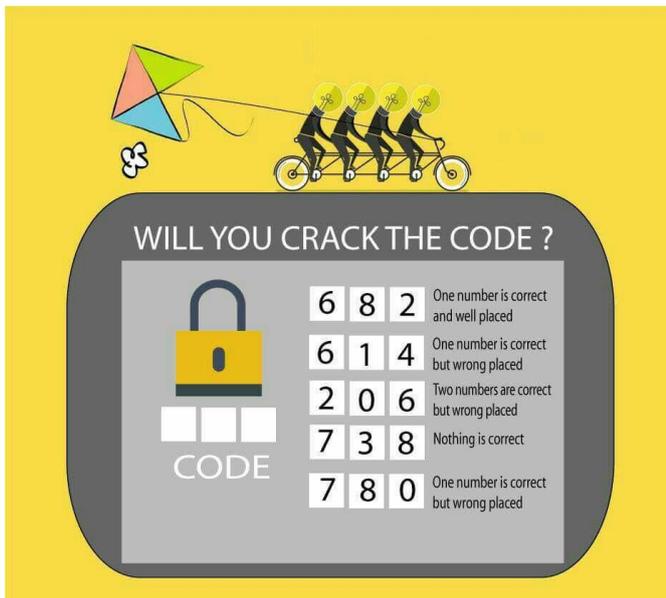
Algoritma brute force adalah pendekatan yang lempeng atau straightforward untuk memecahkan suatu persoalan. Biasanya didasarkan pada pernyataan pada persoalan dan definisi konsep yang dilibatkan. Algoritma brute force memecahkan persoalan dengan sangat sederhana, langsung, dan jelas. Algoritma brute force memiliki kata kunci “Just solve it!”. Ciri-ciri brute force antara lain peningkatan kompleksitas yang tinggi seiring dengan bertambahnya variable pada persoalan, solusi yang dihasilkan pastilah optimum global, dapat menyelesaikan hampir semua persoalan, dan tidak terikat pada beberapa kategori persoalan saja.

Teknik yang digunakan saat menyelesaikan persoalan menggunakan algoritma brute force hanya satu yaitu mencoba segala kemungkinan yang ada, kemudian membandingkan setiap kemungkinan solusi tersebut hingga menemukan solusi terbaik.

Dikutip dari freecodecamp.org mengenai algoritma brute force, terdapat beberapa syarat yang harus dipenuhi suatu algoritma. Syarat-syarat tersebut adalah definiteness, effective computability, dan finiteness. Definiteness yaitu setiap Langkah dinyatakan dengan tepat. Effective computability yaitu setiap Langkah dapat dikerjakan oleh computer. Finiteness yaitu program akan berhenti di suatu saat.

B. Lock Code Puzzle

Lock code puzzle adalah salah satu jenis puzzle matematika. Tujuan dari puzzle ini adalah menemukan permutasi tiga angka yang tepat untuk dapat membuka kunci. Untuk mendapatkan tiga angka yang tepat, pemain diberikan sejumlah petunjuk berupa permutasi angka dan suatu keterangan misalnya banyaknya angka yang tepat atau tidak ada angka yang tepat dan sebagainya. Contoh lock code puzzle adalah berikut



Gambar 1.

Sumber: <https://puzzling.stackexchange.com/questions/46871/crack-the-lock-code>

C. Ragnarok Online

Ragnarok Online, atau sering kita kenal dengan sebutan RO, adalah Massive Multiplayer Online Role Playing Game atau MMORPG yang diciptakan oleh developer game asal Korea Selatan yaitu Gravity Co., Ltd. Game ini diadaptasi dari manhwa dengan judul sama karya Lee Myung-jin. Game ini pertama kali dirilis di Korea Selatan pada tanggal 31 Agustus 2001 untuk sistem operasi Microsoft. Pemain hidup di suatu lingkungan yang terus berubah mengikuti waktu, perubahan-perubahan yang terjadi ditandai dengan episode di lini waktu Ragnarok.

Kata Ragnarok sendiri berasal dari mitologi Norse yang berarti peperangandi akhir zaman. Tidak heran apabila pensuasanaan di game ini banyak mengambil mitologi Norse. Seiring perkembangan game, banyak ditambahkan budaya-budaya lain yang cukup terkenal seperti kebudayaan Jepang kuno yang direpresentasikan oleh kota Amatsu, kebudayaan Cina kuno yang direpresentasikan oleh kota Louyang, kebudayaan Timur Tengah yang direpresentasikan oleh kota Morocc dan lain sebagainya.

Pada awalnya job yang tersedia hanya 13, tapi kini telah berkembang hingga lebih dari 50 job melalui beberapa pembaruan. Gameplay yang ditawarkan oleh game ini pun cukup bervariasi. Pemain bisa melakukan hunting seperti game RPG pada umumnya yang berkonsep PvM (Player vs Monster). Pemain bisa melakukan hunt sendiri untuk mendapat experience point, ataupun menerima quest dari Non-Playable Character (NPC). Bila melakukan quest, pemain bisa mendapatkan hadiah tambahan seperti item maupun skill baru. Terkadang item yang didapat adalah item langka yang tidak bisa didapatkan dari membunuh monster. Ada juga mode PvP (Player versus Player) yang memungkinkan pertarungan antarpemain. Mode ini bisa diakses dengan masuk ke area

khusus PvP yang biasanya ada di tiap kota. Saat PvP, pemain akan ditransfer ke tempat kota yang menyediakan PvP, namun tanpa jalan keluar ataupun NPC, disana pemain bisa mengeluarkan skillnya dan bertarung dengan pemain lain yang ada disana. Namun di beberapa versi RO tertentu disediakan pula server khusus untuk PvP atau PK server, dimana pemain bisa langsung melakukan PvP dimana saja. Namun di server ini dibatasi pemain yang bisa melakukan PvP adalah yang telah berlevel diatas 31. Ada juga event khusus WoE atau War Of Emperium. Event ini adalah salah satu event yang sangat diminati oleh para pemain. Ragnarok Online mulai masuk ke pasar game Indonesia pada bulan Oktober 2003. Sejak kemunculannya di Indonesia, game ini sudah menarik perhatian banyak gamer. Ragnarok juga merupakan salah satu game yang memelopori kepopuleran game online di Indonesia.

Sejak kemunculan pertamanya, Ragnarok Online kini telah memasuki season post 17 Recap Episode. Salah satu fitur yang akan dibahas di makalah ini adalah pembaruan pada Episode 10.3 pada tanggal 5 Oktober 2005, yaitu penambahan Thanatos Tower.



Gambar 2.

D. Lock Code Puzzle di Ragnarok Online

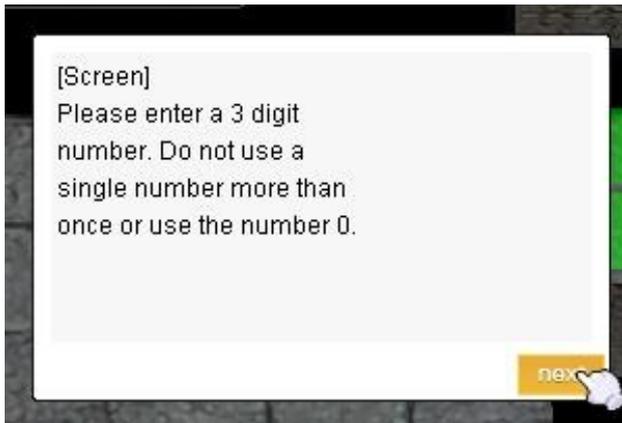
Lock code puzzle di Ragnarok Online ada di dungeon Thanatos Tower. Untuk dapat memasuki Thanatos Tower tidak ada persyaratan khusus bagi pemain. Akan tetapi, untuk memainkan lock code puzzle terdapat satu persyaratan yang harus dipenuhi yaitu terdapat lima pemain online yang telah dikontrak oleh NPC pada lantai satu dan telah berbicara dengan NPC pada lantai dua. Persyaratan tersebut harus dipenuhi untuk membuka warp menuju lantai tiga.

Lock code puzzle berada di lantai empat Thanatos Tower.



Gambar 3.

Untuk memulai lock code puzzle, klik NPC yang terdapat di gambar tengah. NPC tersebut berada di posisi titik kuning yang terdapat di gambar sebelah kanan. Setelah menekan NPC akan muncul dialog box seperti ini.



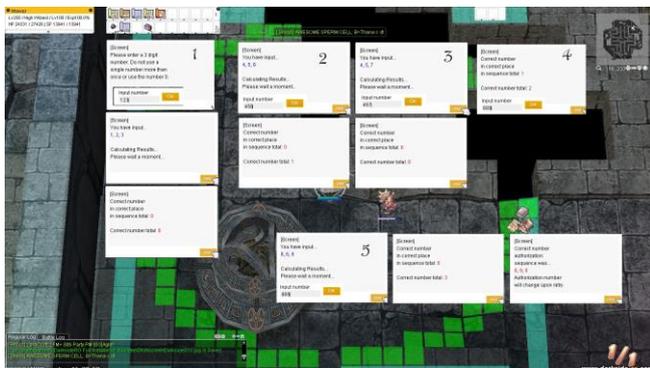
Gambar 4.

III. BATASAN PADA PERSOALAN

Batasan-batasan yang terdapat pada lock code puzzle Thanatos Tower ini antara lain banyaknya percobaan, dan himpunan angka solusi.

A. Banyaknya Percobaan

Batasan ini tidak didefinisikan pada saat memulai permainan. Penulis berhasil mengetahui batasan ini melalui trial and error.



Gambar 5.

Dialog box kanan bawah menunjukkan, setelah percobaan kelima saya, kode unik akan diubah. Itu artinya batasan percobaan adalah lima kali.

B. Himpunan Angka Solusi

Seperti yang terlihat di gambar 4, himpunan solusi merupakan gabungan tiga angka. Angka-angka tersebut tidak berulang dan tidak mengandung nol. Sehingga angka-angka yang mungkin adalah angka 1 hingga angka 9.

Angka yang dimasukkan juga harus berada di posisi yang tepat, sehingga berlaku permutasi. Banyaknya kemungkinannya adalah ${}_9P_3 = 9! / (9-3)! = (9 * 8 * 7 * 6!) / 6! = 9 * 8 * 7 = 504$.

Himpunan angka yang mungkin menjadi solusi:

123	124	125	126
127	128	129	213
214	215	216	217
218	219	...	987

Contoh angka yang tidak mungkin menjadi solusi:

012. Alasan → karena terdapat angka 0.

112. Alasan → karena angka 1 berulang.

C. Mekanisme Permainan

1) Sistem secara acak menentukan tiga angka yang memenuhi batasan B dan harus ditebak oleh pemain.

2) Pemain memasukkan tiga angka yang memenuhi batasan B.

3) Sistem akan memberitahu banyaknya angka tepat yang berada di posisi tepat.

4) Sistem akan memberitahu banyaknya angka tepat yang tidak berada di posisi tepat.

5) Pemain memasukkan tiga angka untuk menebak angka yang ditentukan sistem berdasarkan petunjuk yang telah diberikan pada poin 3 dan 4.

6) Permainan akan terus berlanjut hingga pemain berhasil menebak angka atau pemain telah menebak sebanyak 5 kali.

7) Apabila setelah 5 kali menebak, angka masih belum tepat, sistem akan mengacak angka baru.

IV. ALGORITMA BRUTE FORCE DALAM MENYELESAIKAN LOCK CODE PUZZLE THANATOS TOWER

Terdapat beberapa istilah yang akan digunakan.

1. p_tepat yang menandakan banyaknya angka yang berada di posisi seharusnya.
2. a_tepat yang menandakan banyaknya angka yang tepat tetapi tidak berada di posisi seharusnya.
3. total yang diassign dengan nilai $p_tepat + a_tepat$

A. Algoritma Brute Force Sederhana

Mencoba semua kemungkinan yang ada dengan exhaustive search. Ada sebanyak $9P_3$ kemungkinan.

$$\begin{aligned} 9P_3 &= 9! / (9-3)! \\ &= (9 * 8 * 7 * 6!) / 6! \\ &= 9 * 8 * 7 \\ &= 504 \end{aligned}$$

Kemungkinan terburuk adalah mencoba 504 kali, dan di percobaan ke-504 tebakan angka benar.

B. Ilustrasi Penggunaan Brute Force

Misal angka yang harus ditebak adalah 165

Percobaan 1 = 123

Percobaan 2 = 124

Percobaan 3 = 125

Percobaan 4 = 126

Percobaan 5 = 127

Percobaan 6 = 128

Percobaan 7 = 129

Percobaan 8 = 132

Percobaan 9 = 134

Percobaan 10 = 135

Percobaan 11 = 136

Percobaan 12 = 137

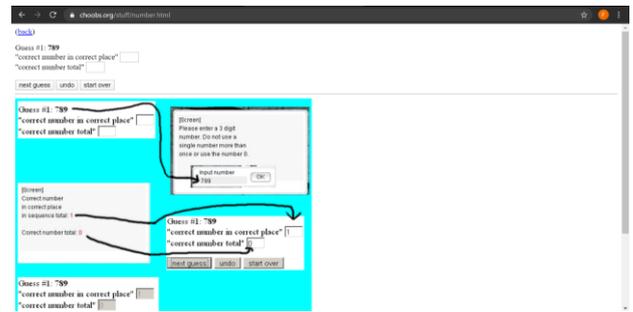
Dst...

Percobaan n = 165

C. Algoritma ...:

Algoritma brute force yang biasa, tidak akan dapat berhasil menyelesaikan lock code puzzle Thanatos Tower karena ada batasan maksimal menebak lima kali sebelum sistem mengacak kunci baru. Penambahan fungsi heuristic ke dalam algoritma memungkinkan persoalan ini diselesaikan dengan maksimal enam kali percobaan.

Algoritma ini diambil dari <https://choobs.org/stuff/number.html>



Langkah-langkah Algoritma

1. Inisialisasi himpunan kandidat untuk setiap digit = [1,2,3,4,5,6,7,8,9], inisialisasi array percobaan = []
2. Coba masukkan 3 angka pertama di himpunan kandidat. Di kasus ini adalah 123. Tambahkan angka yang dicoba ke dalam array percobaan.
3. Masukkan array percobaan sebagai parameter fungsi heuristic. Jika p_tepat = 3, permutasi angka telah ditemukan, program selesai.
4. Fungsi heuristic akan mengembalikan angka untuk dicoba di tahap selanjutnya.
5. Masukkan angka yang dikembalikan oleh fungsi heuristic. Tambahkan angka yang dicoba ke array percobaan.
6. Ulangi kembali dari langkah 3

D. Fungsi Heuristik

Kasus 1:
Misal, kandidat = [1,2,3,4,5,6,7,8,9].
input sebelumnya adalah 123.
p_tepat = 1.
a_tepat = 0.
Fungsi heuristic akan mengembalikan 451.

Kasus 2:
Misal, kandidat = [1,2,3,4,5,6,7,8,9]
Input sebelumnya adalah 123
p_tepat = 1.
a_tepat = 0.
Fungsi heuristic akan mengembalikan 451.
p_tepat = 0.
a_tepat = 0.
1,4,5 akan dihapus dari kandidat.
Fungsi heuristic mengembalikan 627.

E. Ilustrasi Penggunaan Algoritma ...:

Diketahui angka yang akan ditebak adalah 345.

- a. kandidat_belum_coba = [1,2,3,4,5,6,7,8,9]
- b. kandidat untuk digit pertama = []
- c. kandidat untuk digit kedua = []
- d. kandidat untuk digit ketiga = []
- e. array yang menyimpan state percobaan = []

- a. Percobaan pertama: 123
- b. p_tepat = 0
- c. a_tepat = 1
- d. 123, 0, 1 disimpan ke percobaan

- a. panggil fungsi heuristik
- b. fungsi heuristic akan mengembalikan 451. angka 1 diletakkan di posisi terakhir dan mengambil 2 angka dari kandidat_belum_coba.

- a. kandidat_belum_coba = [4,5,6,7,8,9]
- b. kandidat untuk digit pertama = [2,3]
- c. kandidat untuk digit kedua = [1,3]
- d. kandidat untuk digit ketiga = [1,2]
- e. array yang menyimpan state percobaan = [(123,0,1)]

- a. percobaan kedua 451
- b. p_tepat = 0
- c. a_tepat = 2
- d. 451, 0, 2 disimpan ke percobaan

- a. panggil fungsi heuristik
- b. Fungsi heuristic akan mengembalikan 516. 51 diambil dari input sebelumnya karena terdapat dua angka tepat namun posisinya tidak tepat sehingga digeser ke depan.

- a. kandidat_belum_coba = [6,7,8,9]
- b. kandidat untuk digit pertama = [2,3,5]
- c. kandidat untuk digit kedua = [1,3,4]
- d. kandidat untuk digit ketiga = [2,4,5]
- e. array yang menyimpan state percobaan = [(123,0,1), (451,0,2)]

- a. percobaan ketiga 516
- b. p_tepat = 0
- c. a_tepat = 1
- d. 516, 0, 1 disimpan ke percobaan

- a. panggil fungsi heuristic
- b. fungsi heuristic akan mengembalikan 245. 2 diambil dari percobaan pertama karena ada satu angka tepat dan angka tersebut bukan 1. 45 diambil dari percobaan kedua karena ada dua angka tepat

- a. kandidat_belum_coba = [7,8,9]
- b. kandidat untuk digit pertama = [2,3]
- c. kandidat untuk digit kedua = [3,4]
- d. kandidat untuk digit ketiga = [2,4,5]
- e. array yang menyimpan state percobaan = [(123,0,1), (451,0,2), (516,0,1)]

- a. percobaan keempat 245
- b. p_tepat = 2.
- c. a_tepat = 0.
- d. 245, 2, 0 disimpan ke percobaan

- a. panggil fungsi heuristic
- b. fungsi heuristic akan mengembalikan 345. 3 diambil dari percobaan pertama karena ada satu angka tepat dan angka tersebut bukan 1 maupun 2. 45 diambil dari percobaan keempat karena ada dua angka yang posisinya tepat

- a. kandidat_belum_coba = [7,8,9]
- b. kandidat untuk digit pertama = [3]
- c. kandidat untuk digit kedua = [4]
- d. kandidat untuk digit ketiga = [5]
- e. array yang menyimpan state percobaan = [(123,0,1), (451,0,2)]

- a. percobaan kelima 345
- b. p_tepat = 3.
- c. a_tepat = 0.

Terminasi Program
Angka berhasil ditebak

V. KESIMPULAN

Ditulis di web page yang penulis gunakan. Ada beberapa kasus yang tidak dapat diselesaikan hanya dengan lima kali percobaan. Algoritma yang digunakan di web tersebut sangat menarik tetapi mungkin belum diajarkan di semester 4 ini. Akan sangat menarik untuk dibahas jika ada kesempatan lain.

Terima kasih pak Rinaldi Munir yang telah mengajarkan banyak hal selama semester 4 terutama di mata kuliah Strategi Algoritma.

VIDEO LINK YOUTUBE

<https://youtu.be/MPq9VDMrxew>

REFERENSI

- [1] Renouille. 2012. Diakses dari <https://choobs.org/stuff/number.html> pada Sabtu, 3 Mei 2020, pukul 20.26
- [2] Yanuar, Muhamad Rizky. 2009. Diakses dari <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2009-2010/Makalah0910/MakalahStrukdis0910-113.pdf> pada Sabtu, 3 Mei 2020, pukul 11.45
- [3] Munir, Rinaldi. 2014 . Diakses dari <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017->

[2018/Algoritma-Brute-Force-\(2016\).pdf](#) pada Sabtu, 3 Mei 2020 pukul 14.03 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2020



Felix Setiawan 13518078