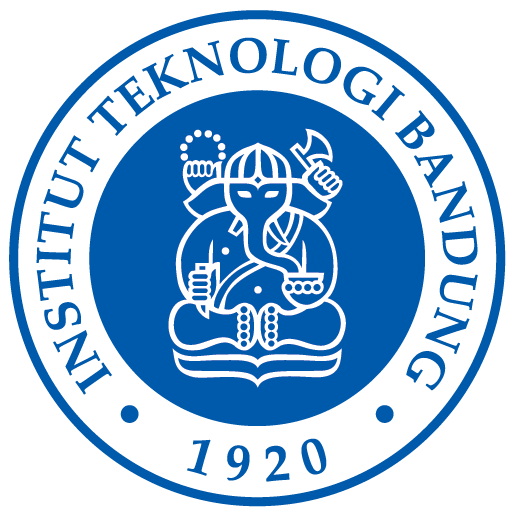
LAPORAN TUGAS KECIL 01

IF2211 STRATEGI ALGORITMA

“Penyelesaian *Cyberpunk 2077 Breach Protocol* dengan Algoritma Brute Force”



Disusun oleh :

Venantius Sean Ardi Nugroho K-02 13522078

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2024

DAFTAR ISI

DAFTAR ISI

BAB 1 DESKRIPSI MASALAH

BAB 2 TEORI SINGKAT

BAB 3 IMPLEMENTASI PROGRAM

BAB 4 EKSPERIMEN

BAB 5 PENUTUP

DAFTAR REFERENSI

BAB 1

DESKRIPSI MASALAH

Pada game *Cyberpunk 2077* yang diciptakan oleh studio *CD Projekt Red* terdapat game yang terinspirasi oleh kegiatan meretas yaitu *Breach Protocol.* Objektif dari game tersebut adalah menemukan kombinasi sekuens yang paling optimal sehingga pemain bisa mendapatkan poin yang terbanyak. Untuk mendapatkan poin, sekuens yang dibuat oleh pemain harus mengandung kombinasi sekuens yang disediakan oleh game. Secara garis besar, berikut adalah komponen – komponen yang terdapat pada game tersebut :

* Token : token adalah kombinasi antara dua karakter alfanumerik yang membangun sebuah sekuens dan matriks
* Matriks : terdiri dar n \* m jumlah token yang akan dipilih untuk menyusun urutan kode
* Sekuens: sebuah rangkaian token (dua atau lebih) yang harus dicocokkan.
* Buffer : jumlah maksimal token yang dapat disusun secara sekuensial.

Untuk membuat kombinasi, pemain harus memilih salah satu token pada posisi baris paling atas dari matriks , lalu bergerak dengan pola vertikal, horizontal, vertikal (bergantian , pasti dimulai dengan vertikal) hingga semua sekuens berhasil dicocokkan atau buffer penuh.

BAB 2

TEORI DAN IDE SOLUSI

Algoritma yang digunakan untuk menemukan sekuens yang optimal dalam permainan peretasan *Cyberpunk 2077* adalah dengan metode *Brute Force*. Algoritma *Brute Force* adalah algoritma yang sederhana dalam menyelesaikan suatu permasalahan, lebih spesifiknya adalah dengan cara mencari semua opsi yang mungkin sampai suatu solusi ditemukan. Ciri – ciri dari algoritma ini adalah algorima *Brute Force* sangat bergantung pada kekuatan komputasi.

Untuk mengerti bagaimana algortima *Brute Force* digunakan dalam menyelesaikan masalah yang diberikan. Alangkah baiknya kita membahas bagaimana alur dari program ini. Berikut merupakan solusi yang saya usulkan untuk memecahkan masalah ini :

1. Meminta data – data yang diperlukan.
2. Menemukan semua kombinasi sekuens sepanjang *buffer* yang dapat dibuat dari matrix lalu menyimpannya dalam sebuah array.
3. Untuk setiap elemen dalam array buffer, akan dicari berapa poin yang didapat dengan cara mengecek apakah suatu sekuens (input) terdapat pada buffer.
4. Masukkan semua point ke dalam array.
5. Temukan point maksimal dalam array tersebut dan ambil indeksnya.
6. Gunakan indeks tersebut untuk menemukan sekuens buffer dengan poin paling banyak.

Dalam alur tersebut, algoritma brute force paling berperan dalam step ke 2 yaitu menemukan semua kombinasi (enumerasi) sekuens yang sepanjang buffer dari matriks. Lebih tepatnya, algoritma *Brute Force* yang digunakan untuk mengenumerasi sekuens adalah algoritma *Backtrack*. Algortima *Back Track* berguna dalam mencari semua solusi dalam suatu permasalahan, biasanya terdapat suatu *constraint* dimana solusi yang tidak memenuhi syaratnya akan dibuang. Algortima *Back Track* yang digunakan menggunakan prinsip *Depth First Search* (DFS). Pada pembuatan sekuen, *constraint* yang perlu dipertimbangkan antara lain adalah :

1. Sekuens tidak boleh memiliki panjang lebih dari *buffer*.
2. Token yang sudah dipakai tidak bisa dipakai lagi.
3. *Sekuens* hanya boleh dibuat dengan arah vertikal – horizontal – vertikal , perlu diingat bahwa gerakan pertama pasti vertikal.
4. Token yang pertama dipilih hanya boleh dari baris paling atas.

Untuk memastikan agar token yang dipakai tidak dipakai tidak digunakn kembali, dibuat sebuah matriks boolean yang dimensinya sama dengan matriks tokennya. Pada implementasi yang saya gunakan, program akan mencoba untuk membuat semua sekuens dengan panjang = buffer, sebelum membuat sekuens dengan panjang 2 <= n < buffer.

BAB 3

IMPLEMENTASI PROGRAM

3.1 *Import statements* dan *classes*

Pada program tucil ini, digunakan library random pada pembuatan sekuens yang digunakan pada opsi input dengan CLI. Lebih spesifiknya, random digunakan untuk menentukan token yang membuat sekuens tersebeut serta poin yang di assign pada tiap sekuens. Library time digunakan untuk menghitung *execution time*.

Dalam pembuatan tugas ini, dibuat class bernama Token dan Matrix. Class Token dibuat untuk memudahkan penyimpanan posisi dari tiap token. Class Matrix dibuat untuk meningkatkan *readability*.

A screenshot of a computer program

Description automatically generated

Gambar 1. *Import Statement* dan *Class* yang dibuat

3.2 Algoritma enumerasi

Berikut adalah kode enumerasi sekuens dengan cara yang telah dijelaskan pada bab 2.



Gambar 2. Fungsi “enumerate\_sequence”

3.3 Algortima Membandingkan Sekuens

Fungsi di bawah ini digunakan untuk menentukan apakah suatu sekuens terdapat dalam suatu sekuens lainnya.

A computer screen shot of a code

Description automatically generated

Gambar 3. Fungsi “compare\_sequence”

3.4 Fungsi *find\_idx*

Fungsi ini dignakan menemukan index suata elemen dalam sebuat array.

A screenshot of a computer program

Description automatically generated

Gambar 4. Fungsi “find\_idx”

3.5 Fungsi *clean\_data*

Fungsi di bawah ini membantu kita dalam mengubah string menjadi elemen – elemen pada sebuah array. Digunakan dalam mengubah inputan txt menjadi array supaya *development* lebih gampang.

A screen shot of a computer program

Description automatically generated

Gambar 5 Fungsi “clean\_data”

3.6 Kamus Global

Kumpulan deklarasi variabel yaang digunakan secara global dalam main.

A screenshot of a computer program

Description automatically generated

Gambar 6 Kamus Global

3.7 Main

3.7.1 Masukkan Data

Terdapat dua cara untuk memasukkan data dalam program, yaitu menggunakan .txt file dan juga melalui inputan CLI. Bila Anda memilih inputan CLI, maka sekuens dan poin dari sekuens tersebut akan dibuat secara random. Bila Anda memilih menggunakan .txt file, pastikan Anda sudah mengikuti format yang ada pada spek.

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

Gambar 7. Penginputan Data pada Program

3.7.2 Alur Utama

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Gambar 8. Snippet Alur Utama

Dafpus

https://www.freecodecamp.org/news/brute-force-algorithms-explained/