LAPORAN

TUGAS BESAR 1 IF2211 STRATEGI ALGORITMA

APLIKASI PERMAINAN KARTU 24 DENGAN ALGORITMA GREEDY

Oleh :

Ferdian Ifkarsyah (13517024)

Harry Rahmadi Munly (13517033)

Nurul Utami Amaliah. W (1351732)

TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2019

BAB I

DESKRIPSI TUGAS

Dalam tugas besar ini, tim wajib merancang dan mengimplementasikan strategi greedy untuk memberikan solusi dalam permainan kartu 24. Karena algoritma greedy membentuk solusi langkah per langkah (step by step), harus ditentukan urutan pemilihan operand, urutan pemilihan operator, dan penggunaan variasi kurung. Tidak boleh menggunakan strategi lain selain greedy.

Fungsi objektif persoalan ini adalah memaksimalkan skor untuk ekspresi solusi yang dihasilkan. Seperti *scrabble*, setiap operator akan memiliki skor. Semakin kompleks operatornya, skor semakin kecil. Skor setiap operator didefinisikan 5 untuk +, 4 untuk -, 3 untuk *, dan 2 untuk /, serta -1 untuk setiap pasang kurung (). Selain skor, operator * dan / memiliki derajat lebih tinggi dibandingkan + dan -, artinya operator berderajat lebih tinggi akan diproses terlebih dahulu. Ekspresi a+b*c-d akan diproses seperti (a+(b*c))-d. Skor akhir ekspresi yang sama dengan 24 tetapi dengan nilai total operator yang lebih rendah akan memiliki bobot yang lebih tinggi daripada ekspresi dengan nilai total operator yang lebih tinggi tetapi tidak menghasilkan nilai 24. Semakin dekat skor akhir ekspresi dengan 24, dengan nilai total operator yang sama, bobotnya akan semakin tinggi.

Lingkungan permainan akan mengeluarkan 4 kartu secara acak. Setiap pemain akan memberikan jawaban masing-masing dan mendapatkan total skor berdasarkan operator yang digunakan. Jika tidak bisa diselesaikan, keempat kartu dikembalikan ke deck dengan urutan acak. Jika pemain memberikan ekspresi yang salah, akan diberikan nilai -10. Permainan diulang sampai dengan dek habis, dan tim pemenang adalah tim dengan skor tertinggi. Aplikasi pemain tidak diperbolehkan membuat cache solusi dari kombinasi supaya lebih cepat, karena akan dibandingkan waktu eksekusi dari implementasi strategi greedy.

Spesifikasi pada tugas besar ini adalah sebagai berikut:

1. Terdapat satu backend engine, dan dua front end:
    a. Backend menghasilkan ekspresi solusi berdasarkan masukan 4 angka.
    b. Front-end pertama berupa GUI yang mendemokan proses pengambilan 4 kartu untuk memberikan input, dan menampilkan hasilnya. Visualisasi kartu untuk demo boleh menggunakan library.
    c. Front-end kedua membaca file masukan, memproses 4 angka dari file masukan, dan menghasilkan file keluaran. Front-end kedua akan berinteraksi dengan lingkungan permainan untuk kompetisi antar tim.
2. Terdapat satu lingkungan permainan yang akan disiapkan oleh asisten untuk kompetisi, yang akan memanggil sejumlah program 24game solver dari tim pemain yang berbeda. Lingkungan permainan ini memiliki engine yang mengatur pemilihan 4 kartu secara acak, memberi masukan ke setiap program

pemain, menerima solusi dari setiap program pemain, dan menghitung skor setiap pemain. Proses ini diulangi sampai dek 52 kartu habis. Aturan tambahan permainan adalah:

a. Pemain dengan skor tertinggi akan menang.
b. Pemain yang tidak menggunakan strategi greedy akan didiskualifikasi.
c. Pemain yang melakukan kecurangan juga akan didiskualifikasi.
d. Setiap kelompok hanya memiliki satu pemain.
e. Pemenang akan dikompetisikan kembali dengan tim pemenang lainnya, sampai keluar 3 tim pemenang dari tim semua kelas (K1-K3).

3. Environment akan memanggil keempat program dengan argument nama file input dan nama file output. Contoh:

   "python batchXX_kelompokYY.py AA BB "

   XX dan YY adalah nomor yang akan diberikan kepada setiap kelompok pada saat pengisian sheet kelompok. AA adalah nama file (termasuk ekstensi file) yang harus dibaca program, dan BB adalah nama file untuk menuliskan ekspresi yang dihasilkan. Perhatikan bahwa nama file berupa variabel, bukan nama file sesungguhnya. Jangan melakukan hardcode untuk melakukan operasi pada file "AA" dan "BB".

   a. File input terdiri dari 1 baris berisi 4 buah integer yang dipisahkan oleh whitespace. Integer memiliki domain [1..13] inklusif.
   b. File output berupa 1 baris ekspresi matematika yang diminta sesuai spek.

4. Strategi *greedy* yang diimplementasikan tiap kelompok harus mempertimbangkan fungsi objektifnya yaitu berusaha memaksimalkan skor. Strategi ini harus dituliskan secara eksplisit pada laporan, karena akan diperiksa pada saat demo apakah strategi yang dituliskan sesuai dengan yang diimplementasikan. Tiap kelompok dapat menggunakan kreativitas mereka dalam menyusun strategi *greedy* untuk memenangkan permainan.
5. Implementasi pemain *24game solver* harus dapat dijalankan pada lingkungan permainan yang telah disediakan oleh asisten.

BAB II

DASAR TEORI

A. Permainan Kartu 24

Dalam permainan kartu 24, terdapat dek (tumpukan) 52 kartu remi. Permainan akan memilih 4 kartu secara acak, lalu setiap pemain akan mencari solusi 24 game dari ke-4 kartu tersebut. Kartu yang diambil tidak dikembalikan lagi ke dek. Nilai yang mungkin dari sebuah kartu adalah 1 (as), 2, …, 10, 11 (jack), 12 (queen), dan 13 (king). Operator yang dapat dipilih + - * / ( ), dan hasil akhir sedekat mungkin dengan nilai 24. Selisih nilai ekspresi solusi dengan 24 akan menjadi pengurang.

B. Algoritma Greedy

Algoritma greedy membentuk solusi langkah per langkah (*step by step*). Pendekatan yang digunakan dalam algoritma Greedy adalah mengambil pilihan "terbaik" dalam setiap langkah yang diambil sebagai optimum lokal. Keputusan yang telah diambil dalam suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Dari pilihan optimum setiap langkah akan diambil solusi optimum global untuk permasalahan.

Algoritma *greedy* adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah:

1. Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (Prinsip "*Take what you can get now!*")
2. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Persoalan optimasi dalam algoritma *Greedy* disusun oleh elemen-elemen seperti, himpunan kandidat (C), himpunan solusi (S), fungsi seleksi, fungsi kelayakan (feasibility), dan fungsi obyektif.

Rumusan algoritma Greedy secara umum, yaitu:

1. Inisialisasi S dengan kosong,
2. Pilih sebuah kandidat (dengan fungsi SELEKSI) dari C,
3. Kurangi C dengan kandidat yang sudah dipilih dari langkah (2) di atas,
4. Periksa apakah kandidat yang dipilih tersebut bersama-sama dengan himpunan solusi membentuk solusi yang layak atau feasible (dilakukan oleh fungsi LAYAK). Jika ya, masukkan kandidat tersebut ke dalam himpunan solusi; jika tidak, buang kandidat tersebut dan tidak perlu dipertimbangkan lagi.
5. Periksa apakah himpunan solusi sudah memberikan solusi yang lengkap (dengan menggunakar-r fungsi SOLUSI. Jika ya, berhenti (selesai); jika tidak, ulangi lagi dari langkah (2).

BAB III

ALGORITMA GREEDY

Strategi Greedy yang dipakai adalah dengan menggunakan pendekatan angka(secara terurut menurun) dan operator(secara terurut menurun berdasarkan skor dari operator). Algoritma ini diimplementasikan dalam bentuk fungsi yang menerima list yang berisi 4 buah integer dan skor target yang dituju(dalam persoaalan ini, 24). Fungsi tersebut akan mengeluarkan keluarannya dalam bentuk string ekspresi aritmetika. Langkah-langkah algoritmanya dijelaskan secara berikut:

1. Urutkan list input secara terurut mengecil. Misal, list L adalah hasil yang sudah terurut. Maka list **L = [x_1,x_2,x_3,x_4]**, dengan $x_1 \geq x_2 \geq x_3 \geq x_4$.
2. Isi ekspresi sementara, misal **expr,** dengan elemen list pertama, **L[0]** atau **x_1**.
3. Untuk setiap 3 angka berikutnya, lakukan hal berikut:
    a. Sediakan 4 buah kandidat ekspresi yang akan ditambahkan ke ekspresi sementara, yaitu:

        $+x_2,-x_2,*x_2,/x_2$

    b. Keempat calon ekspresi tadi, di-*concat* ke **expr** dan dibandingkan mana yang memberikan skor terbesar sementara.
    c. Sebagai contoh, misalkan x_1=4, x_2=3, maka keempat calon ekspresi berikutnya adalah 4+3, 4-3, 4*3, dan 4/3. Masing-masing perhitungan skor keempat ekspresi tersebut dijelaskan dalam tabel berikut:

Ekspresi	Skor Operator	Selisih 24	Total Skor
4+3	+ 5	16	-11
4-3	- 4	23	-19
4*3	● 3	12	-9
4/3	/ 2	22,7	-20,7

    d. Karena ekspresi yang memberikan skor maksimal adalah **4*3**, maka isi dari variabel **expr** sekarang adalah **4*3.**
    e. Lakukan, langkah yang sama untuk **x_3** dan **x_4**

BAB IV

IMPLEMENTASI DAN PENGUJIAN

A. Implementasi

```
fungsi solve_greedy(L : List Angka);
    // Misalkan indeks pertama adalah 0
    urutkan L secara terurut menurut
    expr ← L[0]
    for i : 1 to 3, lakukan:
        cand = [("+L[i]",5),("-L[i]",4),("*L[i]",3),("/L[i]",2)]
        for i : 1 to 4, lakukan:
          exPlus[i] = expr+" + "+cand[i][0]+"+ "+cand[i][1] + "-" 24
        exOptimum = abs(min(eval(setiap exPlus-24) ))
        expr += exOptimum
    → expr
```

B. Pengujian

Untuk melakukan pengujian seberapa efektif dan efisien kah strategi greedy yang digunakan berjalan, kami membandingkannya dengan algoritma brute force yang sudah dipastikan selalu menghasilkan ekspresi yang memberikan skor optimal. Input yang akan digunakan adalah semua kemungkinan kartu mulai dari [1,1,1,1],[1,1,1,2], sampai [13,13,13,13] yang berjumlah 1820 kasus(setiap permutasi dihitung satu kasus).

Kriteria	Greedy	BruteForce
Total Skor	15654.135353535354	22210.349733599734
Skor Benar, = 24	41(2,25%)	410(22,5%)
Skor Salah, /= 24	1779(97,75%)	1410(77,5%)

C. Analisis

Berdasarkan perhitungan skor di atas, algoritma greedy mendapatkan skor total 15654 dari total kemungkinan skor optimal 22210, atau dengan kata lain memiliki skor keoptimalan 70,48%. Namun, jika melihat perbandingan skor yang benar sama dengan 24-nya, algoritma greedy yang kami buat hanya memiliki skor keoptimalan 10%.

Algoritma Greedy yang digunakan gagal dalam 2 jenis kasus, yaitu:

1. Saat menggunakan pengurungan.

Misalnya untuk input [1,1,1,8], algoritma Greedy memberikan ekspresi “8+1+1+1”, sedangkan algoritma Brute Force memberikan ekspresi “(1+1+1)*8”. Hal, ini terjadi karena algoritma Greedy tidak mempertimbangkan kemungkinan pengurungan.

2. Saat penambahan pada input ke-3 sudah sangat besar melewati 24.

   Misalnya, untuk input [13,13,13,13]. Ekspresi sementara sampai bilangan kedua adalah “13+13”. Jika ditambahkan 13 lagi, ekspresi menjadi “13+13+13” dan skor menjadi 10-(39-24)=-5. Sedangkan, ekspresi “13+13-13” memberikan skor 9-(25-13)=-3. Karena algoritma Greedy hanya dapat melihat angka yang tertera saat ini, maka ekspresi ketiga yang dipilih adalah “13+13-13” yang memberikan hasil akhir ekspresi “13+13-13/13”. Padahal, ekspresi optimal yang diberikan algoritma Brute Force adalah “13+13+13-13”

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

- Algoritma Greedy yang dibentuk menghasilkan keoptimalan sekitar 70% dibandingkan dengan Algoritma Brute Force yang sudah pasti akan memberikan hasil optimal.
- Algoritma Greedy menghasilkan nilai 24, 10 kali lebih sedikit dibandingkan Algoritma Brute Force.
- Algoritma Greedy salah pada kasus yang menggunakan pengurungan karena pengurungan tidak menjadi faktor pertimbangan.
- Algoritma Greedy salah pada kasus dimana penambahan bilangan ketiga menghasilkan hasil sementara yang terlalu jauh dari 24.

B. Saran

Sebaiknya, setiap poin spesifikasi yang diberikan, dibuat sejelas-jelasnya. Misalnya, jika ada pembobotan skor yang akan dilakukan jika semakin mendekati 24, diberikan juga skor bobot tersebut secara eksak. Jika boleh, mungkin dapat diberikan juga beberapa contoh ekspresi(misal “10+(11+12)/13”) yang dihasilkan beserta beberapa

DAFTAR PUSTAKA

Sumber tertulis

http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Tugas-Besar-1-IF2211-Strategi-Algoritma-2019.pdf

Munir, Rinaldi. 2007. *Diktat Kuliah IF2211 Strategi Algoritma*. Bandung: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.