UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara
Jujur Adil Wibawa Amanah Religius Akuntabel

Algoritma dan Pemrograman

#12 Meeting

Metode Greedy

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 19900520202406100 1

Definisi

Algoritma Greedy adalah metode pemecahan masalah yang mengikuti heuristik pemecahan masalah dengan membuat pilihan lokal yang optimal pada setiap tahap, dengan harapan bahwa pilihan-pilihan tersebut akan mengarah pada solusi global yang optimal.

Sesuai namanya (Greedy = Tamak/Rakus), algoritma ini selalu ingin mengambil "keuntungan terbesar" yang ada di depan mata saat ini juga, tanpa mempedulikan konsekuensi jangka panjang atau langkah berikutnya.

Prinsip Kerja

Algoritma Greedy bekerja dalam fase-fase. Pada setiap fase:

1. Ia mengambil keputusan yang terlihat terbaik saat ini.
2. Keputusan yang sudah diambil tidak bisa diubah (tidak ada langkah mundur/backtracking).
3. Langkah ini diulang sampai masalah selesai.

Analogi Greedy

Misalkan Anda adalah kasir yang harus memberikan kembalian Rp7.000 menggunakan koin/uang pecahan yang tersedia: Rp5.000, Rp2.000, Rp1.000, Rp500.

- Langkah Greedy: Anda pasti akan mengambil pecahan terbesar yang muat terlebih dahulu.
    a. Ambil Rp5.000 (Sisa Rp2.000).
    b. Ambil Rp2.000 (Sisa 0).
- Anda tidak akan memberikan 7 lembar Rp1.000, meskipun itu mungkin, karena itu tidak efisien (tidak optimal secara jumlah lembar).

Greedy : Optimal Storage on Tapes

Latar Belakang Masalah

Di era modern, kita terbiasa dengan Random Access Memory (RAM) atau SSD di mana mengakses data di posisi manapun membutuhkan waktu yang hampir sama (instan). Namun, dalam sejarah komputasi (dan arsip data besar), kita menggunakan Media Akses Sekuensial seperti pita magnetik (magnetic tapes).

Greedy : Optimal Storage on Tapes

Konsep Kunci: Sequential Access

Bayangkan sebuah kaset lagu lama. Jika Anda ingin mendengarkan lagu ke-5, Anda terpaksa harus memutar (melewati) lagu ke-1, ke-2, ke-3, dan ke-4 terlebih dahulu. Anda tidak bisa langsung "melompat".

- Implikasi: Urutan penyimpanan sangat mempengaruhi seberapa lama kita harus menunggu untuk mendapatkan data.
- Masalah: Jika kita menaruh data yang sering diakses atau data yang pendek di bagian paling belakang, maka "kepala pita" (tape head) harus bekerja keras memutar gulungan pita yang panjang setiap kali mengaksesnya.

Greedy : Optimal Storage on Tapes

Tujuan Optimasi

Tujuannya bukan sekadar "menyimpan data", melainkan meminimalkan Waktu Akses Rata-Rata atau Mean Retrieval Time (MRT).
Kita ingin mencari permutasi (urutan) penyimpanan i(1), i(2), ... , i(n) sedemikian rupa sehingga rata-rata waktu tunggu pengguna menjadi sekecil mungkin.

Greedy : Optimal Storage on Tapes

Logika Greedy pada Optimal Storage on Tapes

- Masalah: Kita ingin mengurangi rata-rata waktu tunggu.
- Sifat "Serakah"-nya: Kita berpikir: "Saya tidak peduli dengan file-file besar nanti. Pokoknya, saya ingin menyelesaikan satu file secepat mungkin sekarang juga agar antrean di belakangnya tidak menunggu lama."
- Keputusan Greedy: Pilih file dengan durasi TERPENDEK untuk ditaruh paling depan.
- Kenapa ini Greedy? Karena kita "serakah" terhadap waktu. Kita ingin "membuang" beban antrean sesegera mungkin.

Greedy : Optimal Storage on Tapes

Analogi Kasir: Bayangkan ada 2 orang di kasir. Orang A bawa 1 permen, Orang B bawa 1 troli penuh belanjaan.

- Greedy: Kasir melayani Orang A dulu (yang pendek). Selesai dalam 10 detik. Orang B menunggu 10 detik + waktunya sendiri. Rata-rata tunggu sangat sebentar.
- Non-Greedy: Kasir melayani Orang B dulu. Orang A harus menunggu 10 menit hanya untuk bayar permen. Rata-rata tunggu jadi lama sekali.

Greedy tidak bekerja secara real time tapi secara Bacth.

Greedy : Optimal Storage on Tapes

Study Cases :

Terdapat 3 program dengan panjang masing-masing:

- Program A: 5 menit
- Program B: 10 menit
- Program C: 3 menit

Greedy : Optimal Storage on Tapes

Skenario 1: Urutan Sembarang (A, B, C)

Urutan penyimpanan: [A] -> [B] -> [C]

1. Akses A: 5 menit.
2. Akses B: 5 (A) + 10 (B) = 15 menit.
3. Akses C: 5 (A) + 10 (B) + 3 (C) = 18 menit.

- Total Waktu: 5 + 15 + 18 = 38 menit.
- MRT: 38 / 3 = 12.66 menit.

Greedy : Optimal Storage on Tapes

Skenario 2: Pendekatan Greedy (C, A, B)

Urutan penyimpanan (Terurut Kecil ke Besar): [C] -> [A] -> [B]

1. Akses C: 3 menit.
2. Akses A: 3 (C) + 5 (A) = 8 menit.
3. Akses B: 3 (C) + 5 (A) + 10 (B) = 18 menit.
   - Total Waktu: 3 + 8 + 18 = 29 menit.
   - MRT: 29 / 3 = 9.66 menit.

Greedy : Knapsack Problem

Latar Belakang Masalah

Ini adalah salah satu masalah optimasi paling terkenal dalam ilmu komputer dan matematika. Masalah ini merepresentasikan situasi Alokasi Sumber Daya Terbatas. Bayangkan Anda memiliki wadah dengan kapasitas terbatas, dan banyak pilihan objek untuk dimasukkan. Setiap objek punya "biaya" (berat/ruang) dan "keuntungan" (nilai/harga). Anda tidak bisa membawa semuanya. Mana yang harus dipilih?

Greedy : Knapsack Problem

Konsep Kunci :

1. 0/1 Knapsack (Biner):
   - Barang tidak bisa dibagi. Anda harus mengambilnya utuh (1) atau tidak sama sekali (0).
   - Contoh: Laptop, Emas Batangan, Lukisan.
   - Solusi: Algoritma Greedy sering gagal di sini. Harus menggunakan Dynamic Programming.
2. Fractional Knapsack (Pecahan):
   - Barang bisa dibagi/dipecah. Anda boleh mengambil sebagian saja (misal: 0.5 kg).
   - Contoh: Gula pasir, Tepung, Biji kopi, Emas bubuk.
   - Solusi: Algoritma Greedy sangat efektif dan optimal untuk kasus ini.

Greedy : Knapsack Problem

Tujuan Optimasi

Tujuannya adalah Maksimasi Profit (Keuntungan).
Kita ingin mengisi ransel sedemikian rupa sehingga total nilai (p) setinggi mungkin, dengan syarat total berat (w) tidak melebihi kapasitas (M).

Greedy : Knapsack Problem

Logika Greedy pada Fractional Knapsack

Dalam Fractional Knapsack, strategi Greedy bukan melihat "Mana yang paling mahal?" atau "Mana yang paling ringan?", tetapi "Mana yang paling berharga per kilogram-nya?".
Ini disebut Densitas Nilai:

Densitas = Profit / Weight

Barang dengan densitas tertinggi adalah prioritas utama untuk dimasukkan ke dalam ransel.

Greedy : Knapsack Problem

Logika Greedy pada Fractional Knapsack

- Masalah: Kita ingin keuntungan maksimal dengan ruang terbatas.
- Sifat "Serakah"-nya: Kita berpikir: "Saya tidak peduli total beratnya berapa, pokoknya saya ingin setiap kilogram yang masuk ke tas saya memberikan nilai rupiah tertinggi."
- Keputusan Greedy: Pilih barang dengan DENSITAS TERTINGGI (Nilai/Berat) untuk diambil duluan.
- Kenapa ini Greedy? Karena kita "serakah" terhadap nilai per unit. Kita ingin setiap jengkal ruang di tas diisi oleh benda paling berharga.

Greedy : Knapsack Problem

Analogi: "Makan di Restoran All You Can Eat (AYCE) Sultan"

Bayangkan Anda ditraktir makan di restoran hotel bintang 5 yang sangat mewah. Namun, ada satu kendala: Perut Anda kapasitasnya terbatas (misalnya cuma muat 500 gram makanan).

Di meja prasmanan tersaji:

1. Nasi Goreng: (Berat, mengenyangkan, tapi murah).
2. Air Mineral: (Berat, tidak ada harganya).
3. Daging Wagyu A5: (Beratnya sama, tapi sangat mahal/enak).
4. Lobster: (Mahal dan enak).

Greedy : Knapsack Problem

Greedy yang Benar (Knapsack Logic): Anda berpikir: "Setiap sendok yang masuk ke mulut saya harus memberikan kenikmatan (nilai) tertinggi."

- Anda akan mengabaikan nasi goreng dan air.
- Anda akan menyerbu Wagyu dan Lobster terlebih dahulu.
- Kenapa? Karena Wagyu memiliki Densitas Kenikmatan Tertinggi. Walaupun cuma makan sepotong kecil (sebagian/fractional), nilainya jauh lebih tinggi daripada sepiring nasi.

Di restoran AYCE, Anda tidak harus memakan satu ekor sapi utuh. Anda bisa mengambil seiris (sebagian) daging Wagyu → inilah sisi Fractional nya.

Greedy : Knapsack Problem

Study Cases :

- Kapasitas Ransel (M): 20 kg
- Daftar Barang:

Barang	Berat (w)	Nilai (p)
A	10 kg	60
B	20 kg	100
C	30 kg	120

Greedy : Knapsack Problem

Langkah Penyelesaian

1. Hitung Densitas (p/w)
   - Barang A: 60 / 10 = 6 (Peringkat 1)
   - Barang B: 100 / 20 = 5 (Peringkat 2)
   - Barang C: 120 / 30 = 4 (Peringkat 3)

2. Proses Seleksi (Iterasi)
   - Iterasi 1 (Barang A):
     - Berat 10 kg <= Sisa Kapasitas 20 kg.
     - Keputusan: Ambil seluruh Barang A.
     - Sisa Kapasitas: 20 - 10 = 10 kg.
     - Nilai: 60.
   - Iterasi 2 (Barang B):
     - Berat 20 kg > Sisa Kapasitas 10 kg.
     - Keputusan: Ambil sebagian Barang B sebesar sisa kapasitas.
     - Fraksi yang diambil: 10 (sisa) / 20 (berat asli)
     - Nilai dari fraksi: 0.5 x 100 = 50.
     - Sisa Kapasitas: 0 kg (Penuh).

3. Hasil Akhir

Total Nilai Maksimum = 60 (Barang A)+50 *Barang B)=110

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara
Jujur Adil Wibawa Amanah Religius Akuntabel

SEE YOU NEXT WEEK !

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 19900520202406100 1
ferdian.bangkit@untirta.ac.id