



UNIVERSITAS BUANA PERJUANGAN KARAWANG
FAKULTAS TEKNIK DAN ILMU KOMPUTER
Terakreditasi BAN-PT

Jl. H.S. Ronggowaluyo Telukjambe Timur Karawang 41361 Telp./Fax. (0267) 8403140
Site: <http://ftik.ubpkarawang.ac.id> email: ftik@ubpkarawang.ac.id

Nama : Muhammad Gusti Nurfathin

NIM : 20416255201133

Kelas : IF20B

Keunggulan dan Kelemahan dari Brute Force

Keunggulan brute-force:

- Sangat sederhana dan mudah untuk dibuat, tidak perlu memerlukan pengetahuan khusus apapun untuk menggunakan atau membuat algoritma jenis ini.
- Dapat digunakan untuk menyelesaikan sebagian besar masalah.
- Sering digunakan untuk menyelesaikan beberapa masalah seperti pencarian, pengurutan, pencocokan string, dan lain-lain.
- Dapat dijadikan sebagai pembandingan terhadap algoritma lain.

Kelemahan brute-force:

- Umumnya tidak efisien dalam menyelesaikan masalah yang besar.
- Tidak cocok untuk memecahkan masalah yang memiliki struktur hierarkis dan melibatkan operasi logis.

Karakteristik Brute Force

- Mudah diimplementasikan.
- Melibatkan perhitungan langsung berdasarkan pernyataan masalah dan definisi dari konsep yang terlibat.
- Mencari semua kemungkinan solusi yang cocok untuk masalah secara keseluruhan.
- Umumnya lambat dan tidak efisien. Tidak cocok untuk menyelesaikan masalah yang besar.
- Memiliki kompleksitas waktu yang besar.

Penyelesaian Kasus Brute Force

Sebuah studio musik membuka layanan sewa studio untuk berlatih musik. Penyewa yang hendak berlatih harus melakukan pendaftaran dua hari sebelum menggunakan studio. Studio hanya membuka layanan selama 14 jam.

Setiap penyewa harus menulis waktu mulai dan waktu berakhir (semua dalam bilangan bulat). Karena permintaan sewa sangat banyak, sementara studio hanya melayani satu penyewa, maka pemilik studio harus memilih dan menjadwalkan grup band yang akan menyewa. Sehingga sebanyak mungkin grup band yang dapat dilayani. Terlampir tabel 10 penyewa.

Tabel 1: Daftar Penyewa

No. Grup Band	1	2	3	4	5	6	7	8	9	10
Mulai	1	3	2	4	8	7	9	11	9	12
Akhir	3	4	5	7	9	10	11	12	13	14

Jika persoalan diselesaikan dengan algoritma Brute Force, bagaimana caranya dan berapa kompleksitas algoritmanya dalam notasi Big O?

Jawaban

Ketentuan Studio hanya dapat melayani satu grup band (penyewa) dalam rentang waktu sewanya dengan batas waktu pelayanan, yang berarti grup band tidak dapat memakai studio jika rentang waktunya bertabrakan dengan rentang waktu grup band lain. Jika bertabrakan atau melebihi batas waktu pelayanan maka sesi grup band tersebut akan dipindahkan ke hari berikutnya. Sebaliknya, grup band akan diizinkan berlatih pada hari tersebut jika rentang waktunya tidak bertabrakan atau melebihi batas waktu pelayanan.

Berikut ini bentuk dari algoritma brute-force berdasarkan ketentuan di atas:

Algorithm 1: Algoritma brute-force penjadwalan penyewa studio

Input: Waktu mulai (s), Waktu selesai (f), Batas waktu (m)

Output: Grup band hari pertama (A), Grup band hari berikutnya (B)

```
1 begin
2    $n \leftarrow \text{len}(s)$  ; // Mengambil panjang input  $s$ 
3    $t \leftarrow 1$  ; // Waktu terakhir pemakaian studio
4    $A \leftarrow \{\emptyset\}$  ; // Jadwal grup band hari pertama
5    $B \leftarrow \{\emptyset\}$  ; // Jadwal grup band hari berikutnya
6   if  $n \neq \text{len}(f)$  then
7     exit
8   // Lakukan proses brute-force
9   for  $i \leftarrow 1$  to  $n$  do
10    if  $s_i \geq t \wedge f_i \leq m$  then
11       $A \leftarrow A \cup \{i\}$ ;
12       $t \leftarrow f_i$ ;
13    else
14       $B \leftarrow B \cup \{i\}$ ;
15  return  $A, B$ 
```

Proses penyelesaian kasus

No. Grup Band	1	2	3	4	5	6	7	8	9	10
Mulai (s)	1	3	2	4	8	7	9	11	9	12
Akhir (f)	3	4	5	7	9	10	11	12	13	14

Memulai dari $i = 1$ dengan $t = 1$ dan $m = 14$

- $i = 1, s_1 = 1, f_1 = 3, t = 1$.

Kondisi $s_1 \geq t \wedge f_1 \leq m$ terpenuhi, maka jadwalkan grup band no. 1 di hari pertama.

Ubah $t \leftarrow f_1$ dan lanjut ke $i = 2$

- $i = 2, s_2 = 3, f_2 = 4, t = 3$.
Kondisi $s_2 \geq t \wedge f_2 \leq m$ terpenuhi, maka jadwalkan grup band no. 2 di hari pertama.
Ubah $t \leftarrow f_2$ dan lanjut ke $i = 3$
- $i = 3, s_3 = 2, f_3 = 5, t = 4$.
Kondisi $s_3 \geq t \wedge f_3 \leq m$ tidak terpenuhi, maka jadwalkan grup band no. 3 di hari berikutnya.
Lanjut ke $i = 4$
- $i = 4, s_4 = 4, f_4 = 7, t = 4$.
Kondisi $s_4 \geq t \wedge f_4 \leq m$ terpenuhi, maka jadwalkan grup band no. 4 di hari pertama.
Ubah $t \leftarrow f_4$ dan lanjut ke $i = 5$
- $i = 5, s_5 = 8, f_5 = 9, t = 7$.
Kondisi $s_5 \geq t \wedge f_5 \leq m$ terpenuhi, maka jadwalkan grup band no. 5 di hari pertama.
Ubah $t \leftarrow f_5$ dan lanjut ke $i = 6$
- $i = 6, s_6 = 7, f_6 = 10, t = 9$.
Kondisi $s_6 \geq t \wedge f_6 \leq m$ tidak terpenuhi, maka jadwalkan grup band no. 6 di hari berikutnya.
Lanjut ke $i = 7$
- $i = 7, s_7 = 9, f_7 = 11, t = 9$.
Kondisi $s_7 \geq t \wedge f_7 \leq m$ terpenuhi, maka jadwalkan grup band no. 7 di hari pertama.
Ubah $t \leftarrow f_7$ dan lanjut ke $i = 8$
- $i = 8, s_8 = 11, f_8 = 12, t = 11$.
Kondisi $s_8 \geq t \wedge f_8 \leq m$ terpenuhi, maka jadwalkan grup band no. 8 di hari pertama.
Ubah $t \leftarrow f_8$ dan lanjut ke $i = 9$
- $i = 9, s_9 = 9, f_9 = 13, t = 12$.
Kondisi $s_9 \geq t \wedge f_9 \leq m$ tidak terpenuhi, maka jadwalkan grup band no. 9 di hari berikutnya.
Lanjut ke $i = 10$
- $i = 10, s_{10} = 12, f_{10} = 14, t = 12$.
Kondisi $s_{10} \geq t \wedge f_{10} \leq m$ terpenuhi, maka jadwalkan grup band no. 10 di hari pertama.

Hasil penjadwalan grup band berdasarkan penyelesaian di atas:

Jadwal grup band hari pertama $\Rightarrow 1, 2, 4, 5, 7, 8, 10$

Jadwal grup band hari berikutnya $\Rightarrow 3, 6, 9$

Kompleksitas Waktu

Algoritma di atas memiliki kompleksitas waktu $O(n)$. Pertumbuhan algoritma ini meningkat secara relatif seiring bertambahnya input.

Implementasi pada Bahasa Pemrograman Python

```
1 def penjadwalan(s, f, m):
2     n = len(s)
3     t = 1
4     A = []
5     B = []
6
7     if n != len(f):
8         raise Exception("Jumlah input waktu mulai dan waktu akhir harus sama")
9
10    for i in range(0, n):
11        if s[i] >= t and f[i] <= m:
12            A.append(i + 1)
13            t = f[i]
14        else:
15            B.append(i + 1)
16
17    return (A, B)
18
19 def main():
20     s = input("Masukkan waktu mulai: ").split()
21     f = input("Masukkan waktu akhir: ").split()
22     m = input("Masukkan batas waktu pelayanan: ")
23     s = [int(i) for i in s]
24     f = [int(i) for i in f]
25
26     try:
27         jadwal = penjadwalan(s, f, int(m))
28
29         print("\nJadwal grup band hari pertama:")
30         for g in jadwal[0]:
31             print("Grup Band No. ", g)
32
33         print("\nJadwal grup band hari berikutnya:")
34         for g in jadwal[1]:
35             print("Grup Band No. ", g)
36
37     except Exception as e:
38         raise e
39
40
41 if __name__ == '__main__':
42     main()
```

Output:

```
1 Masukkan waktu mulai: 1 3 2 4 8 7 9 11 9 12
2 Masukkan waktu akhir: 3 4 5 7 9 10 11 12 13 14
3 Masukkan batas waktu pelayanan: 14
4
5 Jadwal grup band hari pertama:
6 Grup Band No. 1
7 Grup Band No. 2
8 Grup Band No. 4
```

```
9 Grup Band No. 5
10 Grup Band No. 7
11 Grup Band No. 8
12 Grup Band No. 10
13
14 Jadwal grup band hari berikutnya:
15 Grup Band No. 3
16 Grup Band No. 6
17 Grup Band No. 9
```

Source code dokumen ini dan implementasi dapat diambil dari github saya: <https://github.com/gusti32/tugas6-daa>