ASSIGNMENT 1: ALGORITMA BRUTE FORCE DAN REKURSIF

deadline: 2 minggu, sesuai dengan e-learning

Aturan pengerjaan tugas:

1. Kerjakan secara berkelompok (setiap kelompok terdiri dari 3 orang, dan tidak boleh sama dengan kelompok sebelumnya).
2. Lembar ini terdiri dari 3 soal terkait algoritma brute force dan 3 soal terkait algoritma rekursif. Kerjakan **semua** soal yang ada.
3. Bagilah soal berikut dengan kelompok Anda (setiap anggota kelompok wajib membahas satu soal pada bagian I dan satu soal pada bagian II). Setiap anggota kelompok **wajib memahami dan memeriksa pekerjaan dari anggota kelompok lain**, karena ada unsur penilaian kelompok.
4. Buatlah **sebuah** video yang menjelaskan jawaban Anda (dipresentasikan oleh anggota kelompok terkait), namun dalam 1 video. Unggah video Anda di channel Youtube Anda. Pastikan suara dan penjelasan Anda terdengar dengan baik dan jelas. Video tanpa suara yang jelas tidak diperiksa.
5. Anda diizinkan untuk berdiskusi dengan sesama anggota kelompok Anda, tapi tidak diizinkan untuk meminta jawaban ke kelompok lain. Anda dilarang keras mencari jawaban di internet, apalagi menggunakan AI generator. Jika ditemukan kecurangan dalam hal ini, dianggap tidak mengumpulkan tugas.
6. Kumpulkan hasil pekerjaan Anda beserta link video Anda pada link yang tersedia di e-learning. Kumpulkan juga **hardcopy** pada deadline tersebut di ruang tekraf kampus tengah (dikumpulkan per kelas).
7. Format penamaan tugas: **Ass-03_Kelompok_Kelas_NIM1_NIM2_NIM3**
   Contoh: **Ass-03_03_4A_1610101001_1610101002_1610101003**

Penilaian:

- Bobot kelompok: 30% (diambil dari rata-rata nilai anggota kelompok)
- Bobot individu: 70%
- Nilai total: 30% nilai_kelompok + 70% nilai_individu

Dengan ini, Anda menyatakan bahwa Anda siap menerima segala konsekuensi jika nantinya ditemukan adanya kecurangan dalam pengerjaan tugas ini.

Bagian I: Algoritma Brute Force

1. **Travelling Salesman Problem**

   Hamiltonian cycle adalah *cycle* (sirkuit) yang mengunjungi setiap simpul pada graf tepat satu kali. Diberikan sebuah graf lengkap dengan n simpul (graf lengkap adalah graf dengan sifat bahwa untuk setiap pasang simpulnya, terdapat isi yang menghubungkan keduanya), dengan setiap sisinya diberikan bobot berupa bilangan bulat positif. Hal ini dapat dipandang sebagai sekumpulan kota yang terhubung satu sama lain dengan jalur langsung, dan kita tahu jarak antar dua kota.

   a) Buatlah sebuah algoritma brute-force untuk "menemukan sirkuit Hamilton dengan bobot minimum".
   b) Periksalah kebenaran dari algoritma yang Anda rancang.
   c) Buatlah sebuah graf lengkap dengan 5 titik, kemudian ilustrasikan algoritma Anda pada graf tersebut.
   d) Hitunglah kompleksitas waktu algoritma rancangan Anda.
   e) Apakah algoritma yang Anda rancang dapat dimodifikasi untuk diterapkan pada graf input yang <u>bukan</u> merupakan graf lengkap? Jelaskan!

2. **Assignment problem**

   Diberikan n staf dan n tugas. Setiap staff diberikan sebuah tugas. Staff (s_i) ditugaskan ke tugas (t_j) dengan biaya $c(i, j)$.

   a) Mengapa algoritma dibutuhkan untuk menyelesaikan permasalahan ini?
   b) Rancang algoritma brute force untuk menetapkan tugas sedemikian rupa sehingga total biaya $\sum c(i, j)$ diminimalkan.
   c) Periksalah kebenaran dari algoritma yang Anda rancang.
   d) Diberikan 4 staf dan 4 tugas (*assignment*). Buatlah sebuah matriks biaya (dengan mengisikan entri-entri pada matriks di bawah). Terapkan algoritma rancangan Anda untuk menyelesaikan instance permasalahan tersebut.

$$C = \begin{bmatrix} task\ 1 & task\ 2 & task\ 3 & task\ 4 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \begin{matrix} staff\ a \\ staff\ b \\ staff\ c \\ staff\ d \end{matrix}$$

3. **1/0 knapsack problem**

   Knapsack problem didefinisikan sebagai berikut. Diberikan n objek dan ransel berkapasitas K. Setiap objek i memiliki bobot w_i dan untung p_i. Tujuannya adalah menyeleksi objek sehingga total keuntungan dari objek terpilih maksimal, dan berat total benda tidak melebihi kapasitas ransel.

   a) Apa makna "1/0" pada 1/0 Knapsack Problem?
   b) Buatlah sebuah algoritma brute-force untuk menyeleksi objek agar keuntungan maksimal.
   c) Periksalah kebenaran dari algoritma yang Anda rancang.

d) Diberikan lima objek dan sebuah ransel berkapasitas $K = 20$. Lengkapi tabel berikut untuk menjelaskan karakteristik dari setiap objek.

No.	**Object**	**Weight**	**Profit**
1	·	·	·
2	·	·	·
3	·	·	·
4	·	·	·
5	·	·	·

e) Hitunglah kompleksitas waktu algoritma rancangan Anda.

Bagian II: Algoritma Rekursif

1. **Barisan Fibonacci**

   Ingat kembali definisi dari barisan Fibonacci.

   Barisan Fibonacci adalah barisan bilangan yang setiap elemennya merupakan penjumlahan dari dua bilangan sebelumnya, dimulai dari 0 dan 1. Misalnya, 10 angka pertama dalam deret Fibonacci adalah: $0, 1, 1, 2, 3, 5, 8, 13, 21, 34$.

   a) Bagaimana definisi/formula dari barisan Fibonacci secara rekursif?
   b) Buatlah sebuah algoritma rekursif untuk mengkonstruksi barisan Fibonacci dengan n elemen (n diberikan sebagai input).
   c) Buktikan kebenaran algoritma yang Anda rancang.
   d) Ilustrasikan dengan sebuah contoh alur kerja algoritma tersebut, misal dengan mengambil sebuah nilai $n \geq 10$.
   e) Hitunglah kompleksitas algoritma tersebut.
   f) Implementasikan algoritma tersebut pada bahasa pemrograman favorit Anda, dan coba jalankan program tersebut dengan beberapa input sederhana.
   g) Apakah algoritma Anda dapat diubah menjadi algoritma brute-force? Jika iya, jelaskan bagaimana; dan jika tidak, jelaskan mengapa. (Akan sangat bagus jika Anda dapat membuatkan program untuk algoritma versi brute-force, jika ada penyelesaiannya.)

2. **Menara Hanoi**

   Masalah Menara Hanoi adalah masalah klasik dalam bidang Ilmu Komputer dan Matematika, dimana Anda memiliki tiga tiang dan n cakram dengan ukuran berbeda. Cakram awalnya ditumpuk dalam urutan ukuran yang menurun pada satu tiang, dengan yang terbesar di bawah dan yang terkecil di atas. Tujuannya adalah memindahkan seluruh tumpukan ke tiang yang lain, dengan batasan berikut:

   - Hanya satu cakram yang dapat dipindahkan dalam satu waktu.
   - Setiap gerakan terdiri dari mengambil cakram bagian atas dari salah satu tumpukan dan meletakkannya di atas tumpukan lain atau di tiang kosong.
   - Tidak ada cakram yang dapat ditempatkan di atas cakram yang lebih kecil.

   a) Strategi algoritma manakah yang paling cocok untuk menyelesaikan permasalahan menara Hanoi? Jelaskan jawaban Anda.
   b) Jelaskan apa yang menjadi input dan output dari algoritma penyelesaian yang akan Anda rancang.
   c) Rancanglah sebuah algoritma untuk menyelesaikan permasalahan Menara Hanoi.
   d) Periksalah kebenaran dari algoritma yang Anda rancang.
   e) Ilustrasikan algoritma tersebut untuk memindahkan setumpuk cakram (ambil sebuah contoh dimana banyaknya cakram adalah ≥ 4).
   f) Hitunglah kompleksitas waktu dari algoritma tersebut.
   g) Implementasikan algoritma tersebut pada bahasa pemrograman favorit Anda, dan coba jalankan program tersebut dengan beberapa input sederhana.

h) Apakah algoritma Anda dapat diubah menjadi algoritma brute-force? Jika iya, jelaskan bagaimana; dan jika tidak, jelaskan mengapa. (Akan sangat bagus jika Anda dapat membuatkan program untuk algoritma versi brute-force, jika ada penyelesaiannya.)

3. **Algoritma pencarian biner**

Algoritma pencarian biner bekerja dengan cara berulang kali membagi array yang diurutkan menjadi dua dan membandingkan elemen tengah dengan nilai target. Jika elemen tengah sama dengan nilai target, algoritma mengembalikan indeksnya. Jika elemen tengah lebih besar dari nilai target, algoritma mencari bagian kiri array. Jika elemen tengah kurang dari nilai target, algoritma akan mencari bagian kanan dari array. Algoritma mengulangi proses ini hingga nilai target ditemukan atau array kosong.

a) Jelaskan apa yang menjadi input dan output dari algoritma penyelesaian yang akan Anda rancang.

b) Berikan ide bagaimana algoritma tersebut dapat dinyatakan secara rekursif?

c) Periksalah kebenaran algoritma Anda.

d) Berikan sebuah array berukuran 2^n dimana $n \geq 4$, dan ilustrasikan algoritma pencarian biner rancangan Anda dengan menggunakan contoh tersebut.

e) Hitunglah kompleksitas waktu dari algoritma tersebut.

f) Implementasikan algoritma tersebut pada bahasa pemrograman favorit Anda, dan coba jalankan program tersebut dengan beberapa input sederhana.

g) Apakah algoritma Anda dapat diubah menjadi algoritma brute-force? Jika iya, jelaskan bagaimana; dan jika tidak, jelaskan mengapa. (Akan sangat bagus jika Anda dapat membuatkan program untuk algoritma versi brute-force, jika ada penyelesaiannya.)