

UK 1 Strategi Algoritma Kelas A-C-J-I

Poin total **94/98** ?

Email responden (**2200018401@webmail.uad.ac.id**) dicatat saat formulir ini dikirimkan.

0 dari 0 poin

KELAS *

- ☒ A
- ☐ C
- ☐ J
- ☐ I

Nama *

Mohammad Farid Hendianto

NIM *

2200018401

PILIH LAH SALAH SATU JAWABAN YANG ANDA ANGGAP BENAR

94 dari 98 poin



✓ Apa yang dimaksud dengan algoritma brute force? *

4/4

- ☐ a) Algoritma yang menggunakan kecerdasan buatan untuk menyelesaikan masalah
- ☐ b) Algoritma yang mencari solusi optimal secara langsung
- ☒ c) Algoritma yang mencoba semua kemungkinan solusi secara sistematis ✓
- ☐ d) Algoritma yang hanya dapat digunakan untuk masalah kecil

✓ Ketika menggunakan algoritma brute force, apa yang terjadi jika jumlah kemungkinan solusi sangat besar? *4/4

- ☐ a) Algoritma akan memberikan solusi optimal
- ☐ b) Algoritma akan berhenti dan tidak memberikan solusi
- ☒ c) Algoritma akan memakan waktu yang sangat lama untuk mencari solusi ✓
- ☐ d) Algoritma akan memberikan solusi yang salah

✓ Apa kelemahan utama dari algoritma brute force? *

4/4

- ☐ a) Memerlukan perangkat keras yang kuat
- ☐ b) Memerlukan pemrogram yang sangat ahli
- ☒ c) Memerlukan waktu yang sangat lama untuk menyelesaikan masalah dengan ukuran yang besar ✓
- ☐ d) Tidak dapat menyelesaikan masalah apapun



✓ Bagaimana cara mengidentifikasi apakah algoritma brute force merupakan solusi yang tepat untuk suatu masalah?

*4/4

- ☐ a) Dengan mencoba algoritma brute force terlebih dahulu
- ☒ b) Dengan menganalisis kompleksitas waktu algoritma brute force ✓
- ☐ c) Dengan memilih algoritma brute force jika tidak ada solusi yang lain
- ☐ d) Dengan menggunakan algoritma brute force untuk setiap masalah

✓ Kapan algoritma brute force biasanya digunakan? *

4/4

- ☐ a) Ketika solusi optimal diperlukan dengan cepat
- ☐ b) Ketika masalah memiliki jumlah kemungkinan solusi yang terbatas
- ☐ c) Ketika tidak ada algoritma yang lain yang dapat digunakan
- ☒ d) Ketika memerlukan solusi untuk masalah dengan jumlah kemungkinan solusi yang kecil ✓

✓ Apa perbedaan utama antara algoritma brute force dan greedy? *

4/4

- ☒ a) Algoritma brute force mencari solusi optimal, sedangkan greedy mencari solusi yang cukup baik ✓
- ☐ b) Algoritma greedy mencoba semua kemungkinan solusi, sedangkan brute force hanya mencoba solusi yang terdekat
- ☐ c) Algoritma brute force lebih cepat daripada greedy
- ☐ d) Tidak ada perbedaan antara keduanya



✓ Bagaimana kompleksitas waktu dari algoritma brute force dan greedy? * 4/4

- ☐ a) Brute force memiliki kompleksitas waktu yang lebih rendah daripada greedy
- ☒ b) Greedy memiliki kompleksitas waktu yang lebih rendah daripada brute force ✓
- ☐ c) Keduanya memiliki kompleksitas waktu yang sama
- ☐ d) Tergantung pada masalah yang diselesaikan

✓ Ketika memilih solusi, algoritma brute force akan: * 4/4

- ☒ a) Memilih solusi yang optimal pada setiap langkah ✓
- ☐ b) Memilih solusi yang terlihat paling baik pada setiap langkah
- ☐ c) Memilih solusi yang paling sederhana pada setiap langkah
- ☐ d) Memilih solusi secara acak pada setiap langkah

✓ Apa yang dilakukan oleh algoritma greedy ketika memilih solusi? * 4/4

- ☐ a) Menimbang semua kemungkinan solusi
- ☐ b) Memilih solusi yang paling optimal secara global pada setiap langkah
- ☒ c) Memilih solusi yang paling baik secara lokal pada setiap langkah ✓
- ☐ d) Memilih solusi yang paling kompleks pada setiap langkah



✗ Algoritma brute force lebih cocok digunakan ketika: *

0/4

- ☐ a) Masalah memiliki jumlah kemungkinan solusi yang sangat besar
- ☒ b) Masalah memiliki jumlah kemungkinan solusi yang terbatas ✗
- ☐ c) Masalah hanya memiliki satu solusi yang mungkin
- ☐ d) Masalah memiliki solusi yang tidak terdefinisi

Jawaban yang benar

- ☒ a) Masalah memiliki jumlah kemungkinan solusi yang sangat besar

✓ Dalam hal kecepatan, algoritma greedy biasanya lebih cepat daripada brute force karena:

*4/4

- ☐ a) Greedy hanya mempertimbangkan solusi yang terbaik pada setiap langkah
- ☐ b) Greedy tidak perlu melakukan backtracking
- ☐ c) Greedy tidak perlu mempertimbangkan semua kemungkinan solusi
- ☒ d) Semua jawaban di atas benar ✓

✓ Algoritma greedy cenderung menghasilkan solusi yang optimal global? *

4/4

- ☐ a) Ya, karena selalu memilih solusi terbaik pada setiap langkah
- ☒ b) Tidak, karena hanya mempertimbangkan keuntungan lokal pada setiap langkah ✓
- ☐ c) Tergantung pada kompleksitas masalah
- ☐ d) Hanya dalam beberapa kasus khusus



✓ Dalam hal kompleksitas ruang (space complexity), algoritma brute force biasanya: *4/4

- ☐ a) Memiliki kompleksitas ruang yang lebih rendah daripada greedy
- ☒ b) Memiliki kompleksitas ruang yang lebih tinggi daripada greedy ✓
- ☐ c) Keduanya memiliki kompleksitas ruang yang sama
- ☐ d) Tergantung pada implementasi masing-masing algoritma

✓ Kapan algoritma greedy biasanya digunakan? * 4/4

- ☐ a) Ketika solusi optimal diperlukan
- ☐ b) Ketika masalah memiliki jumlah kemungkinan solusi yang besar
- ☒ c) Ketika solusi yang cukup baik sudah memadai ✓
- ☐ d) Ketika masalah memerlukan semua kemungkinan solusi untuk dijelajahi

✓ Algoritma brute force sering digunakan dalam konteks: * 4/4

- ☐ a) Kriptografi
- ☐ b) Perutean (Routing)
- ☐ c) Optimasi fungsi matematika
- ☒ d) Semua jawaban di atas benar ✓



✓ Apa yang menjadi fokus utama dari algoritma Divide and Conquer? * 4/4

- ☐ a) Mengurangi masalah menjadi ukuran yang lebih kecil pada setiap iterasi
- ☒ b) Memecah masalah menjadi submasalah yang lebih kecil, menyelesaikan setiap submasalah, dan menggabungkan solusinya ✓
- ☐ c) Mengurangi kompleksitas waktu dengan mengurangi jumlah langkah yang diperlukan untuk menyelesaikan masalah
- ☐ d) Tidak ada perbedaan antara algoritma Divide and Conquer dengan Decrease and Conquer

✓ Apa prinsip dasar dari algoritma Decrease and Conquer? * 4/4

- ☐ a) Membagi masalah menjadi submasalah yang lebih kecil, menyelesaikan setiap submasalah, dan menggabungkan solusinya
- ☐ b) Mengurangi masalah menjadi ukuran yang lebih kecil pada setiap iterasi
- ☒ c) Mencari pola di dalam masalah dan mengurangi ukurannya secara berurutan ✓
- ☐ d) Tidak ada perbedaan antara keduanya, keduanya mengacu pada prinsip yang sama

✓ Pada tahap mana algoritma Divide and Conquer membagi masalah menjadi submasalah yang lebih kecil? *4/4

- ☒ a) Tahap "divide" ✓
- ☐ b) Tahap "conquer"
- ☐ c) Tahap "combine"
- ☐ d) Tahap "merge"



✓ Pada tahap mana algoritma Decrease and Conquer mengurangi masalah *4/4 menjadi ukuran yang lebih kecil?

- ☐ a) Tahap "divide"
- ☐ b) Tahap "conquer"
- ☒ c) Tahap "reduce" ✓
- ☐ d) Tahap "merge"

✓ Algoritma Decrease and Conquer cenderung menyelesaikan masalah dengan cara *4/4

- ☐ a) Memecah masalah menjadi submasalah yang lebih kecil
- ☐ b) Mengurangi masalah menjadi ukuran yang lebih kecil pada setiap iterasi
- ☐ c) Menggabungkan solusi dari submasalah yang lebih kecil
- ☒ d) Menyelesaikan masalah secara langsung tanpa membaginya ✓

✓ Algoritma Divide and Conquer lebih cocok digunakan untuk masalah yang *4/4 memiliki:

- ☒ a) Struktur hierarkis dan dapat didekomposisi ke dalam submasalah yang lebih kecil ✓
- ☐ b) Masalah yang dapat dipecah menjadi bagian-bagian yang berbeda
- ☐ c) Masalah yang memerlukan pengurangan ukuran pada setiap iterasi
- ☐ d) Masalah yang hanya memiliki satu solusi yang mungkin



✓ Proses reduksi dalam algoritma Decrease and Conquer dilakukan pada tahap: *4/4

- ☐ a) "divide"
- ☐ b) "conquer"
- ☒ c) "reduce" ✓
- ☐ d) "combine"

✓ Dalam algoritma Decrease and Conquer, proses "decrease" mengacu pada: *4/4

- ☐ a) Membagi masalah menjadi submasalah yang lebih kecil
- ☒ b) Mengurangi masalah menjadi ukuran yang lebih kecil pada setiap iterasi ✓
- ☐ c) Menyelesaikan setiap submasalah
- ☐ d) Menggabungkan solusi dari submasalah yang lebih kecil

✓ Algoritma Divide and Conquer sering memiliki kompleksitas waktu yang: * 2/2

- ☐ a) Lebih rendah daripada Decrease and Conquer
- ☒ b) Lebih tinggi daripada Decrease and Conquer ✓
- ☐ c) Sama dengan Decrease and Conquer
- ☐ d) Bergantung pada masalah yang dihadapi



✓ Algoritma Decrease and Conquer cenderung lebih efisien daripada Divide and Conquer dalam kasus-kasus di mana: *4/4

- ☐ a) Masalah dapat dibagi menjadi bagian-bagian yang sama besar
- ☐ b) Masalah memiliki struktur hierarkis yang kompleks
- ☒ c) Masalah memiliki pola yang jelas dan dapat ditempatkan dalam Langkah langkah diskrit ✓
- ☐ d) Tidak ada perbedaan efisiensi antara keduanya

Formulir ini dibuat dalam Universitas Ahmad Dahlan Yogyakarta. [Laporkan Penyalahgunaan](#)

Google Formulir



