

### QUIZ ÔN TẬP

| STT       | Chủ đề                                      | Số câu |
|-----------|---|--------|
| 1         | Độ phức tạp không đệ quy                    | 1      |
| 2         | Độ phức tạp có đệ quy                       | 1      |
| 3         | Kiểm tra hiệu năng bộ test                  | 0      |
| 4         | Brute force, Branch and Bound, Backtracking | 2      |
| 5         | Greedy                                      | 3      |
| 6         | Divide, Decrease and Transform and Conquer  | 2      |
| 7         | Dynamic programming                         | 3      |
| 8         | Thuật toán phân tán                         | 1      |
| 9         | Thuật toán gần đúng                         | 1      |
| Tổng cộng |   | 14     |

**Câu 1: Thuật toán nào dưới đây có độ phức tạp không đệ quy là  $O(n^2)$ ?**

- A. Binary Search
- B. Bubble Sort**
- C. MergeSort
- D. HeapSort

**Câu 2: Độ phức tạp của thuật toán đệ quy sau được xác định bởi phương trình truy hồi  $T(n) = 2T(n/2) + O(n)$ . Độ phức tạp của nó là gì?**

- A.  $O(n)$
- B.  $O(n \log n)$**
- C.  $O(n^2)$
- D.  $O(\log n)$

**Câu 3: Bài toán nào dưới đây thường được giải bằng phương pháp Backtracking?**

- A. Tìm cây khung nhỏ nhất trong đồ thị.
- B. Tìm tất cả các tổ hợp của một tập hợp cho trước**
- C. Sắp xếp mảng bằng thuật toán MergeSort.
- D. Nhân hai số nguyên lớn.

**Câu 4: Khi nào phương pháp Backtracking trở nên không hiệu quả?**

- A. Khi bài toán có ràng buộc cụ thể.
- B. Khi không có cấu trúc con tối ưu.
- C. Khi số lượng trường hợp cần xét quá lớn và không thể cắt tỉa hiệu quả.**
- D. Khi bài toán có thể được chia thành các bài toán con độc lập.

**Câu 5: Bài toán nào dưới đây KHÔNG phù hợp với thuật toán Greedy?**

- A. Tìm cây khung nhỏ nhất của một đồ thị.
- B. Tìm đường đi ngắn nhất từ một đỉnh đến tất cả các đỉnh còn lại trong đồ thị có trọng số dương.**

**C. Tìm dãy con tăng dài nhất trong một mảng.**

D. Phân đoạn lịch trình tối ưu để tối đa hóa số công việc được hoàn thành.

**Câu 6: Trong bài toán “Đường đi ngắn nhất từ một đỉnh đến các đỉnh khác trong đồ thị có trọng số dương”, thuật toán Greedy phù hợp nhất là gì?**

A. Kruskal.

B. Floyd-Warshall.

**C. Dijkstra.**

D. Bellman-Ford

**Câu 7: Đặc điểm nào dưới đây KHÔNG phù hợp với bài toán có thể giải bằng thuật toán Greedy?**

A. Bài toán có cấu trúc con tối ưu (Optimal Substructure).

B. Quyết định tại mỗi bước ảnh hưởng đến kết quả cuối cùng.

C. Mỗi bước chọn phương án tốt nhất tại thời điểm đó.

**D. Lời giải của bài toán lớn phụ thuộc vào lời giải của bài toán con.**

**Câu 8: Đặc điểm nào giúp bạn nhận biết bài toán có thể giải bằng phương pháp Divide and Conquer?**

**A. Kết quả của bài toán lớn được xây dựng từ kết quả của các bài toán con độc lập.**

B. Cần duyệt qua tất cả các trường hợp có thể xảy ra

C. Có thể tối ưu từng bước để đạt được lời giải tốt nhất.

D. Không có trạng thái con lặp lại trong bài toán.

**Câu 9: Thuật toán nào dưới đây là ví dụ điển hình của Decrease and Conquer?**

A. QuickSort

B. Binary Search

**C. Euclid's Algorithm (thuật toán tìm GCD).**

D. Đếm số lần xuất hiện của phần tử trong một dãy.

**Câu 10: Khi nào bài toán thích hợp để áp dụng Dynamic Programming?**

A. Lời giải của bài toán con không được tái sử dụng.

B. Mỗi bước chỉ cần chọn phương án tốt nhất.

**C. Có trạng thái con lặp lại và các bài toán con gộp nhau.**

D. Bài toán không thể chia nhỏ thành các bài toán con.

**Câu 11: Trong bài toán nhân ma trận chuỗi (Matrix Chain Multiplication), phương pháp nào tối ưu nhất để giải?**

A. Divide and Conquer

B. Greedy

**C. Dynamic Programming**

D. Brute Force

**Câu 12: Bài toán nào dưới đây thích hợp với phương pháp Dynamic Programming?**

**A. Tìm số cách đi từ góc trái trên đến góc phải dưới của một ma trận, chỉ được đi xuống hoặc sang phải.**

- B. Tìm cây khung nhỏ nhất của một đồ thị.
- C. Tìm tất cả các dãy con liên tiếp của một mảng.
- D. Tìm dãy con dài nhất của một chuỗi (Longest Common Subsequence).

**Câu 13: Phương pháp nào dưới đây thường được dùng để giải các bài toán tối ưu NP-khó?**

- A. Brute Force
- B. Greedy
- C. Thuật toán gần đúng (Approximation Algorithm).
- D. Divide and Conquer

**Câu 14: Phương pháp nào dưới đây KHÔNG phải là thuật toán phân tán?**

- A. MapReduce
- B. QuickSort
- C. Blockchain Consensus Algorithms
- D. Distributed Hash Tables (DHT)