



# Quiz ôn tập CS112

**Câu 1:** Bạn Quang đã giải được bài toán tháp Hà Nội nhưng bạn lại không biết độ phức tạp của thuật toán trong bài toán này. Hãy giúp Quang biết độ phức tạp về thời gian, về không gian của thuật toán này là bao nhiêu?

- A.  $O(n), O(2n)$
- B.  $O(n), O(2^n)$
- C.  $O(2^n), O(n)$
- D.  $O(2n), O(n)$

```
void TowerOfHanoi(int numberOfDisks, char sourcePeg, char aux.  
    if (numberOfDisks == 1) {  
        cout << "Move disk 1 from " << sourcePeg << " to " <<  
        return;  
    }  
  
    TowerOfHanoi(numberOfDisks - 1, sourcePeg, targetPeg, aux.  
  
    cout << "Move disk " << numberOfDisks << " from " << sour
```

```
TowerOfHanoi(numberOfDisks - 1, auxiliaryPeg, sourcePeg,  
}
```



Đáp án đúng là C

**Câu 2: Phương pháp thiết kế thuật toán nào được sử dụng ở đoạn mã dưới đây?**

- A. Greedy Algorithms
- B. Divide and Conquer
- C. Brute Force
- D. Brand and Bound

```
int count(int a[], int l, int r, int x){  
    if (l == r){  
        if (a[l] == x) return 1;  
        else return 0;  
    } else {  
        int m = (l+r)/2;  
        return count(a, l, m, x) + count(a, m+1, r, x);  
    }  
}
```



Đáp án đúng là B

**Câu 3: Điểm khác biệt giữa Dynamic Programming và Divide and Conquer là gì?**

- A. Các bài toán con trong Dynamic Programming không được giải độc lập
- B. Kết quả của các bài toán con trong Dynamic Programming không được lưu lại.
- C. Dynamic Programming là kết hợp lời giải của các bài toán con để tìm ra lời giải của bài toán ban đầu.

D. Dynamic Programming không sử dụng kết quả của bài toán con.



Đáp án đúng là A

**Câu 4: Dynamic Programming có phải là bài toán tối ưu cho mọi trường hợp không?**

A. True

B. Flase



Đáp án đúng là B

**Câu 5: Giải thuật sắp xếp Quick Sort và Merge Sort sử dụng phương pháp thiết kế thuật toán nào?**

A. Brand and Bound

B. Brute Force

C. Divide and Conquer

D. Dynamic Programming



Đáp án đúng là C

**Câu 6: Tại sao phải tính toán độ phức tạp thuật toán? Hãy chọn câu sai:**

A. Đánh giá hiệu suất

B. So sánh thuật toán

C. Tối ưu hóa

D. Để “professional” hơn



Đáp án đúng là D

**Câu 7: Có bao nhiêu bước để phân tích độ phức tạp của thuật toán đệ quy?**

- A. 6
- B. 4
- C. 7
- D. 5



Đáp án đúng là 5

**Câu 8: Thuật toán nào sau đây KHÔNG phải là thuật toán chia để trị về bản chất?**

- A. Euclidean algorithm to compute the greatest common divisor
- B. Heap Sort
- C. Cooley-Tukey fast Fourier transform
- D. Quick Sort



Đáp án đúng là B

**Câu 9: Thuật toán nào sau đây không phải là thuật toán Backtracking?**

- A. Knight tour problem
- B. N queen problem
- C. Tower of Hanoi
- D. M coloring problem



Đáp án đúng là C

**Câu 10: Phương pháp thiết kế nào không sử dụng Optimal Substructure?**

- A. Greedy
- B. Brute Force
- C. Dynamic Programming
- C. Divide and Conquer



Đáp án đúng là B

**Câu 11: Cho một đa giác lồi  $N$  đỉnh. Bằng các đường chéo không cắt nhau, ta có thể chia đa giác thành  $N - 2$  tam giác. Hãy xác định cách chia có tổng các đường chéo ngắn nhất. Vậy phương pháp thiết kế thuật toán nào phù hợp?**

- A. Branch and Bound
- B. Backtracking
- C. Divide and Conquer
- D. Dynamic Programming



Đáp án đúng là D

**Câu 12: We use dynamic programming approach when**

- A. We need an optimal solution
- B. The solution has optimal substructure
- C. The given problem can be reduced to the 3-SAT problem
- D. It's faster than Greedy



Đáp án đúng là B