socrative

Name _____
Date _____

CS112 - N17: Greedy Algorithm

Score _____

1. Xét về mặt tối ưu, thuật toán tham lam sẽ **đảm bảo** tạo ra được giải pháp tối ưu vì nó **xem xét tất cả trường hợp** có thể xảy ra và sau đó chọn giải pháp tốt nhất. Đúng hay Sai?

(T) True

(F) False

2. Cho **N sự kiện** với thời gian **bắt đầu** và **kết thúc** của chúng, chọn **tối đa M sự kiện** được thực hiện bởi một người (*giả sử một người chỉ thực hiện một hoạt động duy nhất vào một thời điểm*). Chiến lược tham lam nào được coi là hiệu quả?
VD: N = 6
start [i] = { 1, 3, 0, 5, 8, 5 }
end [i] = { 2, 4, 6, 7, 9, 9 }

(A) thời gian kết thúc ít nhất và thời gian bắt đầu nhiều hơn (hoặc bằng) thời gian kết thúc

(B) thời gian kết thúc nhiều nhất và thời gian bắt đầu nhiều hơn (hoặc bằng) thời gian kết thúc

(C) thời gian kết thúc ít nhất và thời gian bắt đầu ít hơn (hoặc bằng) thời gian kết thúc

3. Mệnh đề nào sau đây đúng về thuật toán tham lam?

(A) Chọn giải pháp tối ưu ở từng bước nhằm hướng đến giải pháp tối ưu cho cả vấn đề

(B) Sẽ quay lui lại bước trước khi gặp trường hợp phát sinh không tối ưu ở bước sau.

(C) Chia bài toán thành nhiều bài toán con và tìm giải pháp cho từng bài toán con.

4. Chọn các nhận định đúng về 2 thuật toán **Tham lam** (greedy) và **Vét cạn** (brute force)

(A) Vét cạn đảm bảo kết quả bài toán hơn Tham lam.

(B) Đều cùng chọn giải pháp tốt nhất trong tất cả các trường hợp xảy ra.

(C) Hai thuật toán đều gặp vấn đề về **Độ phức tạp lớn**

(D) Cả hai thuật toán đều đều lựa chọn giải pháp tối ưu ở từng bài toán con

5. Nhận định đúng về thuật toán **Tham lam** và **Chia để trị**

(A) Cả hai đều đảm bảo dẫn đến kết quả tối ưu

(B) **Chia để trị** chia bài toán thành các bài toán con và chọn bài toán con tối ưu để giải quyết

(C) **Cấu trúc con tối ưu** của **Tham lam** thể hiện ở giải pháp tối ưu toàn cục nằm trong giải pháp tối ưu cục bộ

(D) Phân tích ***độ phức tạp*** của **Tham lam** dễ dàng hơn **Chia để trị**

6. Thuật toán tham lam sử dụng **không đảm bảo kết quả** trong bài toán nào dưới đây?

(A) Dijkstra's Shortest Path Algorithm

(B) Minimum number of coins required

(C) Kruskal's Minimum Spanning Tree

(D) Job Sequencing Problem

7. Độ phức tạp của Thuật toán Dijkstra trong tìm đường đi ngắn nhất? (mã giả ảnh)
+ **V** - Đỉnh
+ **E** - Tổng số cạnh

(A) O(V.E.logV)

(B) O(E.logV)

(C) $O(E^V.logV)$

(D) O(V.logE)

dijkstra.py

```
function Dijkstra(Graph, source):
    dist[source] ← 0

    create vertex priority queue Q

    for each vertex v in Graph:
        if v ≠ source
            dist[v] ← INFINITY
            prev[v] ← UNDEFINED

        Q.add_with_priority(v, dist[v])

    while Q is not empty:
        u ← Q.extract_min()
        for each neighbor v of u:
            alt ← dist[u] + length(u, v)
            if alt < dist[v]
                dist[v] ← alt
                prev[v] ← u
                Q.decrease_priority(v, alt)

    return dist, prev
```

8. Bạn cần trả 12 xu.
Bạn có các xu với giá trị là: 1; 3; 6; 10
Sử dụng **thuật toán Tham lam** để tìm số xu tối thiểu để trả.

(A) 6;3;1;1;1

(B) 6;6

(C) 6;3;3

(D) 10;1;1