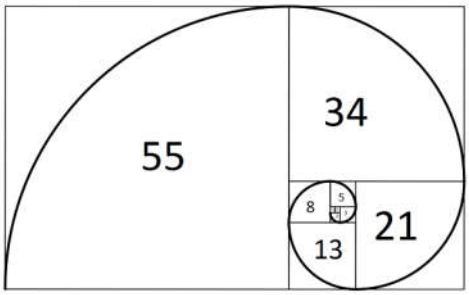
BỘ XÂY DỰNG
TRƯỜNG ĐHXD MIỀN TÂY

ĐÁP ÁN – THANG ĐIỂM
KỲ THI KTHP HỌC KỲ I NĂM HỌC 2022-2023
Trình độ: ĐẠI HỌC; Ngày thi: ……
Môn: Phân Tích và Thiết Kế Thuật Toán

ĐÁP ÁN ĐỀ THI CHÍNH THỨC
(Đáp án - thang điểm gồm 04 trang)

Câu	Đáp án	Nội dung	Thang điểm
I		**TRẮC NGHIỆM**	**5.0**
1	C	Tìm đường đi ngắn nhất	0.25
2	B	Boolean	0.25
3	C	Backtracking	0.25
4	D	Kiểu dữ liệu trừu tượng	0.25
5	A	Độ phức tạp về thời gian và không gian	0.25
6	B	O(n)	0.25
7	C	Merge Sort	0.25
8	B	Merge Sort	0.25
9	C	Stack	0.25
10	A	O(n^2)	0.25
11	D	NP – complete problem	0.25
12	C	O(log2n)	0.25
13	D	O(n^2)	0.25
14	B	Bubble Sort	0.25
15	C	C. T(N) = 2T(N-1)+1	0.25
16	D	Insert Sort	0.25
17	A	O(1)	0.25
18	B	Là dạng mô tả lỗi cú pháp	0.25
19	C	O(N Log N)	0.25
20	A	O(n^2)	0.25
II		**TỰ LUẬN**	**5.0**

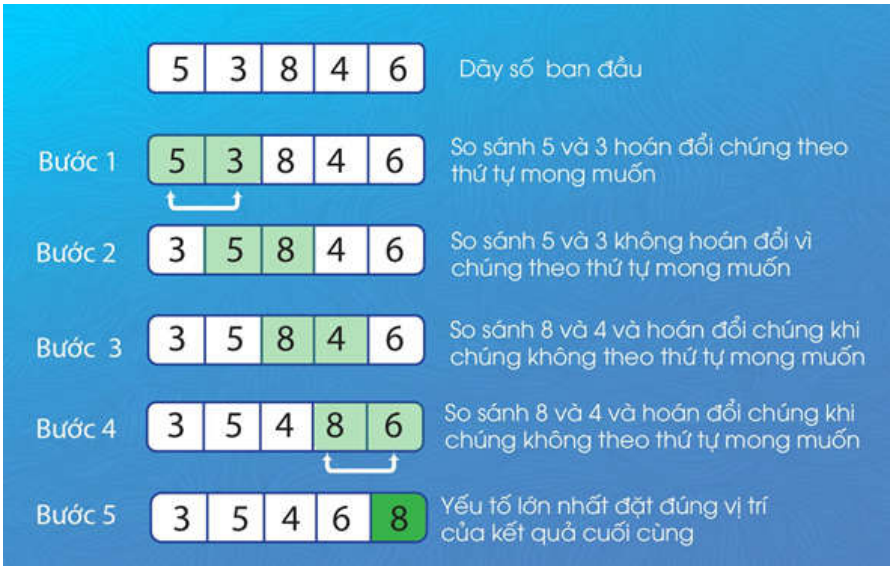
1	Câu 1	Trình bày Thuật toán **Bubble Sort** (Sắp xếp nổi bọt) để sắp xếp một mảng số nguyên, ứng dụng minh họa sắp xếp dãy số: **5, 3, 8, 4, 6**	1.0
2	Câu 2	Dùng ngôn ngữ lập trình **Python** để hiện thực Thuật toán **Bubble Sort**	1.0
3	Câu 3	Dùng ngôn ngữ lập trình **Python** để hiện thực Thuật toán tính dãy số **Fibonacci** thứ N bằng phương pháp quy nạp (đệ quy), viết hàm kiểm tra in 10 số **Fibonacci** đầu tiên 	1.5
4	Câu 4	Dùng Thuật toán **Insertion Sort** thực hiện các yêu cầu sau a) Vẽ lưu đồ Thuật toán (flow chart) (1.0 điểm) b) Phân tích ưu điểm, nhược điểm, thời gian thực thi và tính độ phức tạp của thuật toán (0.5 điểm)	1.5
		Tổng điểm	**10,0đ**

Đáp án (Tự Luận)

1. **Câu 1:**

Xét một mảng gồm **nn** số nguyên: **a1, a2, a3,…, an** và **a1, a2, a3,..., an**

- Với cách sắp xếp không giảm từ trái qua phải, mục đích của chúng ta là đưa dần các số lớn nhất về cuối dãy (ngoài cùng bên phải).
- Bắt đầu từ vị trí số 11, xét lần lượt từng cặp 22 phần tử, nếu phần tử bên phải nhỏ hơn phần tử bên trái, ta sẽ thực hiện đổi chỗ 22 phần tử này, nếu không, xét tiếp cặp tiếp theo. Với cách làm như vậy, phần tử nhỏ hơn sẽ "nổi" lên, còn phần tử lớn hơn sẽ "chìm" dần và về bên phải.
- Khi kết thúc vòng thứ nhất, ta sẽ đưa được phần tử lớn nhất về cuối dãy. Sang vòng thứ hai, ta tiếp tục bắt đầu ở vị trí đầu tiên như vậy và đưa được phần tử lớn thứ hai về vị trí thứ hai ở cuối dãy ...



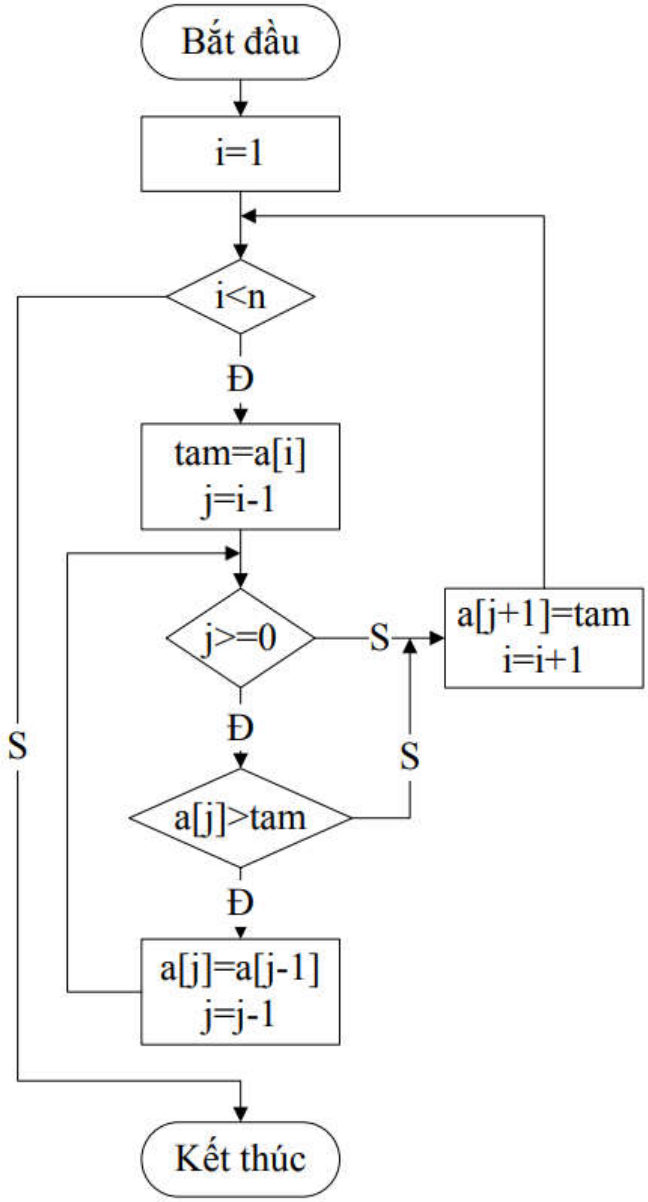
2. **Câu 2:** Dùng ngôn ngữ lập trình **Python** để hiện thực Thuật toán **Bubble Sort**

```python
def bubbleSort(arr):
    n = len(arr)
    swapped = False
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j + 1]:
                swapped = True
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
        if not swapped:
            return
```

3. **Câu 3:** Dùng ngôn ngữ lập trình **Python** để hiện thực Thuật toán tính dãy số **Fibonacci** thứ N bằng phương pháp quy nạp (đệ quy), viết hàm kiểm tra in 10 số **Fibonacci** đầu tiên

```python
def fibonacci(n):
    if (n < 0):
        return -1
    elif (n == 0 or n == 1):
        return n;
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
print("10 số đầu tiên của dãy số fibonacci: ")
sb = ""
for i in range(0, 10):
    sb = sb + str(fibonacci(i)) + ", "
print(sb)
```

4. **Câu 4**
   - **a)** Vẽ lưu đồ Thuật toán (flow chart)



b) Phân tích ưu điểm, nhược điểm, thời gian thực thi và tính độ phức tạp của thuật toán

- Ưu điểm:
  - ✓ Là thuật toán cơ bản, dễ hiểu, phù hợp cho người bắt đầu học về sắp xếp.
  - ✓ Đoạn code ngắn gọn, dễ nhớ
- Nhược điểm:
  - ✓ Hiệu suất chậm nhất trong các thuật toán sắp xếp.
  - ✓ Không hiệu quả với những dữ liệu lớn
- Thời gian thực thi và Độ phức tạp của thuật toán:
  Với mỗi i=1,2,..,n−1i=1,2,..,n−1 ta cần n−in−i phép so sánh. Do đó số nhiều nhất các lần so sánh và đổi chỗ trong giải thuật là: (n−1)+(n−2)+...+2+1 = (n−1)n2(n−1)+(n−2)+...+2+1 = 2(n−1)n Do đó ta có độ phúc tạp là O(n2)O(n2).