**Đề gồm 10 câu.**

**Câu 1:** Nêu ý tưởng thuật toán HeapSort.

**Câu 2:** Độ phức tạp trong tất cả các trường hợp của thuật toán sắp xếp HeapSort là:

A. O(n2)

B. O(n)

C. O(nlog2n)

D. O(log2n)

Đáp án: C

**Câu 3**: Cần bao nhiêu lần hoán vị để đưa giá trị lớn nhất trong dãy Max-Heap sau về cuối mảng: 17, 16, 15, 7, 9, 13, 14, 8.

A. 1

B. 2

C. 3

D. 4

Đáp án: A

**Câu 4**: Thực hiện thuật toán BuildHeap trên dãy sau: 14, 2, 13, 21, 16, 9, 10, 4 , 8, 17. Hỏi cặp liên đới của số 17 trong lần hiệu chỉnh cây Heap đầu tiên là những cặp số nào?

A. 8 và 16

B. 8 và 14

C. 4 và 16

D. 16 và 13

Đáp án: A. Sau khi thực hiện buildHeap ta được cây sau tương ứng:



**Câu 5**: Thực hiện sắp xếp dãy sau bằng thuật toán HeapSort: 14, 2, 13, 21, 16, 9, 10, 4 , 8, 17. Hỏi sau bao nhiêu lần hiện chỉnh thành cây Heap thì dãy được sắp xếp tăng dần. Hãy trình bày từng bước.

Bài giải: Sau 9 lần hiệu chỉnh thành cây Heap thì dãy được sắp xếp. (SV nên trình dày dạng cây)

| Dãy ban đầu | 14 2 13 21 16 9 10 4 8 17 |  |
| --- | --- | --- |
| Bước 1: | 21 17 13 8 16 9 10 4 2 14  14 17 13 8 16 9 10 4 2 21 | Dãy sau khi thực hiện BuildHeap với mảng xét từ chỉ số i=0 -> 9  Swap(a[0], a[9]) |
| Bước 2: | 17 16 13 8 14 9 10 4 2 21  2 16 13 8 14 9 10 4 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 8  Swap(a[0], a[8]) |
| Bước 3: | 16 14 13 8 2 9 10 4 17 21  4 14 13 8 2 9 10 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 7  Swap(a[0], a[7]) |
| Bước 4: | 14 8 13 4 2 9 10 16 17 21  10 8 13 4 2 9 14 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 6  Swap(a[0], a[6]) |
| Bước 5: | 13 8 10 4 2 9 14 16 17 21  9 8 10 4 2 13 14 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 5  Swap(a[0], a[5]) |
| Bước 6: | 10 8 9 4 2 13 14 16 17 21  2 8 9 4 10 13 14 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 4  Swap(a[0], a[4]) |
| Bước 7: | 9 8 2 4 10 13 14 16 17 21  4 8 2 9 10 13 14 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 3  Swap(a[0], a[3]) |
| Bước 8: | 8 4 2 9 10 13 14 16 17 21  2 4 8 9 10 13 14 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 2  Swap(a[0], a[2]) |
| Bước 9: | 4 2 8 9 10 13 14 16 17 21  2 4 8 9 10 13 14 16 17 21 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 1  Swap(a[0], a[1]) |

**Câu 6**: Thực hiện sắp xếp dãy sau bằng thuật toán HeapSort: 7, 9, 10, 15, 8, 5, 6, 1, 4. Hỏi sau bao nhiêu lần hiện chỉnh thành cây Heap thì dãy được sắp xếp giảm dần. Hãy trình bày từng bước.

| Dãy ban đầu | 7, 9, 10, 15, 8, 5, 6, 1, 4 |  |
| --- | --- | --- |
| Bước 1: | 1, 4, 5, 7, 8, 10, 6, 15, 9  9, 4, 5, 7, 8, 10, 6, 15, 1 | Dãy sau khi thực hiện BuildHeap với mảng xét từ chỉ số i=0 -> 8  Swap(a[0], a[8]) |
| Bước 2: | 4, 7, 5, 9, 8, 10, 6, 15, 1  15, 7, 5, 9, 8, 10, 6, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 7  Swap(a[0], a[7]) |
| Bước 3: | 5, 7, 6, 9, 8, 10, 15, 4, 1  15, 7, 6, 9, 8, 10, 5, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 6  Swap(a[0], a[6]) |
| Bước 4: | 6, 7, 10, 9, 8, 15, 5, 4, 1  15, 7, 10, 9, 8, 6, 5, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 5  Swap(a[0], a[5]) |
| Bước 5: | 7, 8, 10, 9, 15, 6, 5, 4, 1  15, 8, 10, 9, 7, 6, 5, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 4  Swap(a[0], a[4]) |
| Bước 6: | 8, 9, 10, 15, 7, 6, 5, 4, 1  15, 9, 10, 8, 7, 6, 5, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 3  Swap(a[0], a[3]) |
| Bước 7: | 9, 15, 10, 8, 7, 6, 5, 4, 1  10, 15, 9, 8, 7, 6, 5, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 2  Swap(a[0], a[2]) |
| Bước 8: | 10, 15, 9, 8, 7, 6, 5, 4, 1  15, 10, 9, 8, 7, 6, 5, 4, 1 | Dãy sau khi Heapify tại chỉ số i=0 với mảng xét từ chỉ số i=0 -> 1  Swap(a[0], a[1]) |

**Câu 7**: Hãy nêu ý tưởng của thuật toán MergeSort.

**Câu 8**: Thực hiện sắp xếp dãy sau bằng thuật toán MergeSort: 14, 2, 13, 21, 16, 9, 10, 4 , 8, 17. Sử dụng 2 phương pháp đệ quy và không đệ quy.

**Câu 9**: Thực hiện sắp xếp dãy sau bằng thuật toán MergeSort: 7, 9, 10, 15, 8, 5, 6, 1, 4. Sử dụng 2 phương pháp đệ quy và không đệ quy.

**Câu 10:** (Code trên máy tính) Viết chương trình sắp xếp mảng ngẫu nhiên trên 10 ngàn phần tử và so sánh thời gian chạy của 2 thuật toán HeapSort và MergeSort.