IDHLI01

– Suppose that you want to sort a singly linked list, each list's item is a large object. which of the following sort algorithms should be used to minimum the time complexity?

Select one:

- Quick sort.
- Insertion sort.
- Heap sort.
- Bubble sort.

IDHLI02

– Method reverse() below is used to reverse the order of items in a Singly Linked List. Please complete the code of the method?

```
int maxSum=0;
for (int i=0; i<a.length; i++)
  for (int j=i; j<a.length; j++)
  {
    int thisSum=0;
    for (int k=i; k<=j; k++)
      thisSum+=a[k];
    if (thisSum>maxSum)
      maxSum=thisSum;
  }
return maxSum;
```

Select one:

- prev=head
- head=current
- current=head
- head=prev

IDHLI03

– Consider a Singly Linked List contains N nodes (N > 8), a method f1() is designed to find the 8th node from beginning, and method f2() is designed to find the 8th node from end. Which is the time complexity of f1() and f2()?

Select one:

- O(N) and O(1)
- O(N) and O(N)
- O(1) and O(N)
- O(1) and O(1)

IDHLI04

– Method deleteTail() below is used to delete the last node in a Singly Linked List. Please complete the code of the method?

```
public void deleteTail()
{
  int pos = getLength();
  SLNode beforeTail=traversing(pos-1);
  __________;
}
```

Select one:

- beforeTail.setNext(null)
- beforeTail = tail
- beforeTail.setNext(tail)
- beforeTail = null

IDHLI05

– Method tailToFront() below moves the last node of a Singly Linked List into the front of the list. Please complete the code of the method?

```
public void tailToFront()
{
  int pos = getLength();
  SLNode beforeTail=traversing(pos-1);
  SLNode tail=beforeTail.getNext();

  ________
}
```

Select one:

- beforeTail.setNext(null); tail.setNext(head); head=tail;
- tail.setNext(null); beforeTail.setNext(head); head=beforeTail;

tail.setNext(head); head.setNext(null); beforeTail=head;

head.setNext(null); beforeTail.setNext(tail); tail=beforeTail;