Dæmi 1:
B,C og D virka ekki þar sem que er “first in – first out” þarf að prenta út 0 fyrst. Þ.e. það er einungis hægt að prenta út rununa 0 1 2 3 4 5 6 7 8 9.

Dæmi 2:

```
import java.util.*;

public class DN
{
  private DoubleNode first = null;
  private DoubleNode last = null;

  private class DoubleNode
  {
    String item;
    DoubleNode next;
    DoubleNode befo;
  }

  public void pushFront(String item)
  {
    DoubleNode oldfirst = first;
    first = new DoubleNode();
    first.item = item;
    first.next = oldfirst;
    first.befo = null;
    oldfirst.befo = first;
  }

  public void pushBack(String item)
  {
    DoubleNode oldlast = last;
    last = new DoubleNode();
    last.item = item;
    last.next = null;
    last.befo = oldlast;
    oldlast.next = last;
  }

  public void removeFirst()
  {
    first = first.next;
    first.befo = null;
  }

  public void removeLast()
  {
    last = last.befo;
    last.next = null;
  }

  public void insertBefo(DoubleNode n)
  {
    DoubleNode in = new DoubleNode();
    DoubleNode prev = new DoubleNode();
    prev.next = in;
```

```
  in.next = n;
  n.befo = in;
  in.befo = prev;
}

public void insertAfter(DoubleNode n)
{
  DoubleNode in = new DoubleNode();
  DoubleNode next = n.next;
  next.befo = in;
  in.befo = n;
  n.next = in;
  in.next = next;
}

public void remove(DoubleNode n)
{
      if (first == null)
      {
          return;
      }
      DoubleNode temp = first;
      if (n == first)
      {
          first = temp.next;
          return;
      }
      DoubleNode prev = n.befo;
      DoubleNode next = prev.next.next;
      prev.next = next;
      next.befo = prev;
}
```

Dæmi 3:

a) Forritið keyrir 2N sinnum, N + 1/2N + 1/4N + … sem stefnir á 2 þegar N hækkar. Svo tímaflækjan er O(N) þar sem 2 er fasti.
b) Forritið keyrir 2N sinnum svo tímaflækjan er O(N).
c) Forritið keyrir Nlog(N) sinnum svo tímaflækjan er O(N).

Dæmi 4:

```
import java.util.Arrays;

public class pair
{
  public static void main(String [] args)
  {
    In in = new In();
    int [] a = in.readAllInts();
    Arrays.sort(a);
    for(int i = 0; i < a.length; i++)
    {
      System.out.println(a[i] + " ");
    }
    int count = 0;
    for(int i = 0; i < a.length - 1; i++)
    {
      int j = i + 1;
      int curr = 1;
      while(j < a.length && a[i] == a[j])
      {
        j++;
        curr++;
      }
      i += curr - 1;
      count += (curr*(curr - 1))/2;
    }
    System.out.println(count);
  }
}
```

Dæmi 5:

```
public static int rank(int [] a, int key)
  {
    int lo  = 0;
    int hi = a.length - 1;
    int mid = 0;
    while (lo <= hi)
    {
      mid = lo + (hi - lo) / 2;
      if(mid == 0 || a[mid] > a[mid - 1])
      {
        return mid;
      }
      else if(key <= a[mid])
      {
        hi = mid - 1;
      }
      else if(key > a[mid])
      {
        lo = mid + 1;
      }
    }
    if(a[mid] == key)
    {
      return mid;
    }
    else
    {
      return -1;
    }
  }
```

Dæmi 6:

```
public class sumThree
{
  public static void main(String [] args)
  {
    for(int N = 100; N <= 6400; N*=2)
    {
      int [] a = new int[N];
      for(int s = 0; s < N; s++)
      {
        a[s] = (int)(Math.random()*100) - 50;
      }
      int cnt = 0;
      long start = System.currentTimeMillis();
      for (int i = 0; i < N; i++)
      {
        for (int j = 0; j < N; j++)
        {
          for (int k = 0; k < N; k++)
          {
            if (i < j && j < k)
            {
              {
                if (a[i] + a[j] + a[k] == 0)
                {
                  cnt++;
                }
              }
            }
          }
        }
      }
      long timi = System.currentTimeMillis() - start;
      System.out.println(N + ": " + timi);
    }
  }
}
```

```
Welcome to DrJava.  Working directory is /Users/sessihers/OneDrive/Tolfr102
> run sumThree
100: 7
200: 21
400: 152
800: 935
1600: 7278
3200: 58905
6400: 480901
```

Svo við fáum að tímaflækjan er O(N^3).