Question 1

In [1]:

```
a = [1,2,3,8,4,3,2]

def find_index1(l):
    # initial key(current max) with the first element in the list
    # initial result(max index) with the first index in the list
    key = l[0]                  # c1*1
    result = 0                  # c2*1
    for i in range(1, len(l)):   # c3*n
        # if current element > current max, update current max and max index
        if a[i]>key:            # c4*(n-1)
            result = i          # c5*(n-1)
            key = l[i]          # c6*(n-1)
    return result               # c7*1

find_index1(a)
```

Out[1]:

3

Initialization

The for loop start from the second elements, before the loop starts, a[0] in the list is the current max value and 0 is the current max index. This is True because so far we only consider one element in the list. Besides, a[0] is still the original element in the list. Therefore, the loop invariant holds prior to the first iteration.

Maintenance

The body of the for loop works by comparing the new/next element in the list with current key(current max value), if the new element is larger, update the key and max index. So after each loop, the key and result still hold the current max value and its index, among all the elements it have checked during the loop. Besides, the original list is not modified in the body of this for loop. So in each iteration, the loop invariant maintains.

Termination

When iteration index is larger than the length of list, which means the iteration comes to the end of the list, the iteration terminates. During the iteration, the algorithm keep track of the largest elements and its index in the list, therefore, when the iteration terminates, we look through all the elements in the list and store the max value in key and store it's index in result. We conclude that the index we find indeed represent the largest value. Hence, the algorithm is correct.

Runing Time

The best case and the worst case are the same:

$$T(n) = c1 + c2 + c3 * n + c4 * (n-1) + c5 * (n-1) + c6 * (n-1) + c7$$
$$= \Theta(n)$$

Question 2

Basis:

$L_0 = 2, L_1 = 1$ then, according to L's formula, when n>1, we have:

$$L_2 = L_1 + L_0 = 3$$

It also satisfy the closed-form expression for the n-th Lucas number:

$$L_2 = (\frac{1+\sqrt{5}}{2})^2 + (1 - \frac{1+\sqrt{5}}{2})^2$$
$$L_2 = \frac{3}{2} + (1+\sqrt{5}) + \frac{3}{2} - (1+\sqrt{5})$$
$$L_2 = 3$$

Induction Hypothesis:

Suppose that for a positive integer k, the following equation holds:

$$L_k = (\frac{1+\sqrt{5}}{2})^k + (1 - \frac{1+\sqrt{5}}{2})^k$$
$$L_{k-1} = (\frac{1+\sqrt{5}}{2})^{k-1} + (1 - \frac{1+\sqrt{5}}{2})^{k-1}$$

Also, by defination we have:

$$L_k = L_{k-1} + L_{k-2}$$

Induction step:

since by definition:

$$L_{k+1} = L_k + L_{k-1}$$

and

$$L_k = (\frac{1+\sqrt{5}}{2})^k + (1 - \frac{1+\sqrt{5}}{2})^k$$
$$L_{k-1} = (\frac{1+\sqrt{5}}{2})^{k-1} + (1 - \frac{1+\sqrt{5}}{2})^{k-1}$$

then for L_{k+1}, we can conclude:

$$L_{k+1} = L_k + L_{k-1}$$

$$L_{k+1} = (\frac{1+\sqrt{5}}{2})^k + (1 - \frac{1+\sqrt{5}}{2})^k + (\frac{1+\sqrt{5}}{2})^{k-1} + (1 - \frac{1+\sqrt{5}}{2})^{k-1}$$

$$L_{k+1} = (\frac{1+\sqrt{5}}{2})^{k-1} \cdot (1 + \frac{1+\sqrt{5}}{2}) + (1 - \frac{1+\sqrt{5}}{2})^{k-1} \cdot (\frac{1-\sqrt{5}}{2} + 1)$$

$$L_{k+1} = (\frac{1+\sqrt{5}}{2})^{k-1} \cdot (\frac{3+\sqrt{5}}{2}) + (1 - \frac{1+\sqrt{5}}{2})^{k-1} \cdot (\frac{3-\sqrt{5}}{2})$$

$$L_{k+1} = (\frac{1+\sqrt{5}}{2})^{k-1} \cdot (\frac{1+\sqrt{5}}{2})^2 + (1 - \frac{1+\sqrt{5}}{2})^{k-1} \cdot (1 - \frac{1+\sqrt{5}}{2})^2$$

$$L_{k+1} = (\frac{1+\sqrt{5}}{2})^{k+1} + (1 - \frac{1+\sqrt{5}}{2})^{k+1}$$

which is the given closed form. so we proved that this closed form is correct.

Question 3.a

To prove Θ is equivalence relation, we need to prove it is reflexive, symmetric and transitive.

Reflexive

Given a $f(n)$, $\exists c_1, c_2, n_0$ such that,

$$\forall n \geq n_0,\ 0 \leq c_1 f(n) \leq f(n) \leq c_2 f(n)$$

So we conclude that Θ is reflexive.

Symmetry

Give that, $f(n) = \Theta(g(n))$ By defination, we have $\exists c_1, c_2, n_0$ such that,

$$\forall n \geq n_0,\ 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

We have the following two inequation by transforming the inequation above,

$$g(n) \leq \frac{f(n)}{c1} \leq \frac{c_2}{c_1} \cdot g(n)$$

$$\frac{c_1}{c_2} \cdot g(n) \leq \frac{f(n)}{c2} \leq g(n)$$

Then we have,

$$\frac{f(n)}{c2} \leq g(n) \leq \frac{f(n)}{c1}$$

So we can rewrite the inequation, $\exists c_3 = \frac{1}{c2}, c_4 = \frac{1}{c1}, n_0$ such that,

$$\forall n \geq n_0,\ 0 \leq c_3 f(n) \leq g(n) \leq c_4 f(n)$$

Which indicates $g(n) = \Theta(f(n))$ and Θ is symmetry.

Transitive

Suppose we have $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$, by defination, we have $\exists c_1, c_2, n_0, c_3, c_4, n_1$ such that,

$$\forall n \geq n_0,\ 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

$$\forall n \geq n_1,\ 0 \leq c_3 h(n) \leq g(n) \leq c_4 h(n)$$

Then, we have,

$$\forall n \geq n_0, n_1,\ 0 \leq c_1 c_3 h(n) \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \leq c_2 c_4 h(n)$$

Which indicates that $f(n) = \Theta(h(n))$, so Θ is transitive.

Overvall, Θ is equivalence relation.

Question 3.b

It's obvious that $\frac{1}{2}(f(n) + g(n)) \leq max(f(n), g(n)) \leq f(n) + g(n)$, since $f(n) > 0,\ g(n) > 0$

We can take $c_1 = \frac{1}{2}$ and $c_2 = 1$, then we have, $\exists c_1 = \frac{1}{2}, c_2 = 1, n_0$ such that,

$$\forall n \geq n_0, c_1(f(n) + g(n)) \leq max(f(n), g(n)) \leq c_2(f(n) + g(n))$$

Which indicates $\max(f(n), g(n)) = \Theta(f(n) + g(n))$. Maximum of two functions is in Θ of their sum.

Question 3.c

In the last Question we have prove that Θ is reflexive. so in question 3.b, we proved $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

By reflexive, it indicates that $f(n) + g(n) = \Theta \max(f(n), g(n))$. Sum of two functions is in Θ of their maximum.

Question 4

Informal statement:

Rank from lowest to highest: constant, logarithmic, linear, linearithmic, polynomial, exponential.

Formal statement:

- Constant and Logarithmic

> Given constant M, $\forall c, \exists n_0 = \frac{e^M}{c}$ such that,
> $$\forall n \geq n_0, 0 \leq C \leq clog(n)$$
> So, $constant = o(log(n))$

- Logarithmic and Linear

> By L'Hôpital's rule, n and $log(n)$ are differentiable at their domin.
> $\lim_{n\to\inf} \frac{log(n)}{n} = \lim_{n\to\inf} \frac{\frac{1}{n}}{1} = 0$ Which means, $log(n) = o(n)$

- Linear and Linearithmic

> By L'Hôpital's rule, n and $nlog(n)$ are differentiable at their domin.
> $\lim_{n\to\inf} \frac{n}{nlog(n)} = \lim_{n\to\inf} \frac{1}{1+log(n)} = 0$ Which means, $n = o(nlog(n))$

- Linearithmic and Polynomial

> By L'Hôpital's rule, $n^{constant}$ and $nlog(n)$ are differentiable at their domin.
> $\lim_{n\to\inf} \frac{nlog(n)}{n^{constant}} = \lim_{n\to\inf} \frac{1+log(n)}{n^c \cdot log(n)} = 0$ Which means, $nlog(n) = o(n^{constant})$

- Polynomial and Exponential

> By L'Hôpital's rule, n^{c1} and c_2^n are differentiable at their domin.
> $\lim_{n\to\inf} \frac{n^{c1}}{c_2^n} = \lim_{n\to\inf} \frac{c1 n^{c1}}{c_2^n \cdot log(c2)}$ Apply L'Hôpital's rule repeatly, we have,
> $\lim_{n\to\inf} \frac{n^{c1}}{c_2^n} = \lim_{n\to\inf} \frac{c1! \cdot 1}{c_2^n \cdot (log(c2)^{c_1})} = 0$ Which means, $n^{c1} = o(c_2^n)$