In [ ]:
```
import numpy as np
import time

np.random.seed = 69420
def generate_seq(n=15, min_val=0, max_val=14):
    return np.random.randint(min_val, max_val, n)

def lcs(A, B, i, j):
    if len(A) <= i or len(B) <= j:
        return 0
    if (A[i] == B[j]):
        return 1 + lcs(A, B, i+1, j+1)

    return max(lcs(A, B, i+1, j), lcs(A, B, i, j+1))
```

In [ ]:
```
for i in range(5,20, 5):
    for j in range(5,20, 5):
        print("----------------------------------------")
        t = time.time()
        A, B = generate_seq(i), generate_seq(j)
        print("lcs:  {}, ({:03.2f} s) A: {:3}, B: {:3}".format(lcs(A, B, 0, 0)
```

```
----------------------------------------
lcs:  1, (0.00 s) A:   5, B:   5
----------------------------------------
lcs:  4, (0.00 s) A:   5, B:  10
----------------------------------------
lcs:  3, (0.01 s) A:   5, B:  15
----------------------------------------
lcs:  1, (0.00 s) A:  10, B:   5
----------------------------------------
lcs:  4, (0.06 s) A:  10, B:  10
----------------------------------------
lcs:  4, (0.39 s) A:  10, B:  15
----------------------------------------
lcs:  3, (0.00 s) A:  15, B:   5
----------------------------------------
lcs:  3, (0.36 s) A:  15, B:  10
----------------------------------------
lcs:  6, (7.14 s) A:  15, B:  15
```

1.

hér eru keyrslur fyrir öll n og m á bilunum [5...15] með 5 staka millibili, tíminn vex með veldisvextinum c^m þar sem c er fasti staka í n

In [ ]:
```
dict = {}
def lcs2(A, B, i, j):
    if len(A) <= i or len(B) <= j:
        return 0
    if (i,j) in dict and dict[(i,j)] == 1:
        return 1 + lcs2(A,B, i+1, j+1)
    elif A[i] == B[j]:
        dict[(i,j)] = 1
        return 1 + lcs2(A,B,i+1,j+1)

    dict[(i,j)] = 0
    return max(lcs2(A,B,i+1,j), lcs2(A,B,i,j+1))
```

In [ ]:
```
for i in range(5,20, 5):
    for j in range(5,20, 5):
        dict = {}
        print("----------------------------------------")
        t = time.time()
        A, B = generate_seq(i), generate_seq(j)
        print("lcs:  {}, ({:03.2f} s) A: {:3}, B: {:3}".format(lcs(A, B, 0, 0)
        t = time.time()
        print("lcs2: {}, ({:03.2f} s) A: {:3}, B: {:3}".format(lcs2(A, B, 0, 0
```

```
----------------------------------------
lcs:  1, (0.00 s) A:   5, B:   5
lcs2: 1, (0.00 s) A:   5, B:   5
----------------------------------------
lcs:  2, (0.00 s) A:   5, B:  10
lcs2: 2, (0.00 s) A:   5, B:  10
----------------------------------------
lcs:  2, (0.01 s) A:   5, B:  15
lcs2: 2, (0.02 s) A:   5, B:  15
----------------------------------------
lcs:  0, (0.00 s) A:  10, B:   5
lcs2: 0, (0.01 s) A:  10, B:   5
----------------------------------------
lcs:  2, (0.07 s) A:  10, B:  10
lcs2: 2, (0.08 s) A:  10, B:  10
----------------------------------------
lcs:  5, (0.19 s) A:  10, B:  15
lcs2: 5, (0.23 s) A:  10, B:  15
----------------------------------------
lcs:  1, (0.01 s) A:  15, B:   5
lcs2: 1, (0.02 s) A:  15, B:   5
----------------------------------------
lcs:  3, (0.77 s) A:  15, B:  10
lcs2: 3, (0.92 s) A:  15, B:  10
----------------------------------------
lcs:  3, (43.01 s) A:  15, B:  15
lcs2: 3, (66.45 s) A:  15, B:  15
```

2.

hér má sjá keyrslur á bæði upprunalega lcs og seinna fallinu sem notar memoisation, það er greinilegt að ekki mikið gagn er að minninu

þetta er vegna þess að fallið hefur ekkert að gera með gömul gildi á `(i,j)` því þessi pör endurtaka sig ekki