End-to-Beginning(sequence/array A):

```
n = sizeof(A) //1
H[n] //1
for(i=0; i<n; i++) //n
    H[i]=0 //1

For(i=n-2; !(i<0); i--): //n-2
    For(j=i+1; j<n; j++): // n-(i+1)+1
        if(A[i] < A[j] && H[i]≤H[j]): //3
            H[i] = H[j]+1 //2
max = maxelementof(H)+1 //2
R[max] //1
r=0 //1
Less = max-1 //2
For(i=0; i<n && r!=max; i++): //n
    if(H[i]==Less) //1 + max(3,0) = 4
        R[r] = A[i] //1
        r++ //1
        Less-- //1
return R //1
```

$+\max(2,0) = 5$

$\Rightarrow \sum_{i=0}^{n-2}(n-i)$

$(n-2)n - \frac{(n-2)(n-1)}{2}$

$(n^2-2n) - \cancel{1}\left(\frac{n^2}{2}-\frac{3n}{2}+\frac{2}{2}\right)$

$\frac{1}{2}n^2 - \frac{1}{2}n - 1$

$]-4n$

$T(n) = n + (\frac{1}{2}n^2 - \frac{1}{2}n - 1)\cdot 5 + 4n + 9$

$T(n) = n + \frac{5}{2}n^2 - \frac{5}{2}n - 5 + 4n + 9$

$T(n) = \frac{5}{2}n^2 + \frac{5}{2}n + 4$

Where $f(n)=n^2$ Prove $T(n) = \Theta(n^2)$ by limit THM. As $L>0$ AND $<\infty$.

$$\lim_{n\to\infty} \frac{T'(n)}{f(n)} = \frac{d/dn\,(\frac{5}{2}n^2+\frac{5}{2}n+4)}{d/dn\,(n^2)} \rightarrow \lim_{n\to\infty} \frac{5n+\frac{5}{2}}{2n} = \cancel{\frac{\infty}{\infty}}$$

$$\lim_{n\to\infty} \frac{d/dn(5n+\frac{5}{2})}{d/dn\,(2n)} = \frac{5}{2} > 0 \text{ AND } \frac{5}{2} < \infty$$

$$\therefore \frac{5}{2}n^2 + \frac{5}{2}n + 4 = \Theta(n^2)$$

$2^2=4$

KB		
00	~~∅~~	0
10	K	1
01	B	1
11	KB	2

$\frac{2^n}{2}$

The idea of creating (generating) every possible subsequences from a sequence

		a b c		0 8 4 12	# ~~of~~ index searching through
~~0~~	∅ =	0 0 0	1	0 0 0 0 = ∅	~~0~~
~~1~~	a =	1 0 0	2	1 0 0 0 = 0	~~1~~
~~1~~	b =	0 1 0	3	0 1 0 0 = 8	~~1~~
~~2~~	ab =	1 1 0	4	1 1 0 0 = 0 8	~~2~~
~~1~~	c =	0 0 1	5	0 0 1 0 = 4	~~1~~
~~2~~	ac =	1 0 1	6	1 0 1 0 = 0 4	~~2~~
~~2~~	bc =	0 1 1	7	0 1 1 0 = 8 4	~~2~~
~~3~~	abc =	1 1 1	8	1 1 1 0 = 0 8 4	3
			9	0 0 0 1 = 12	~~1~~
			10	1 0 0 1 = 0 12	~~2~~
			11	0 1 0 1 = 8 12	~~2~~
			12	1 1 0 1 = 0 8 12	~~3~~
			13	0 0 1 1 = 4 12	~~2~~
			14	1 0 1 1 = 0 4 12	~~3~~
			15	0 1 1 1 = 8 4 2	~~3~~
			16	1 1 1 1 = 0 8 4 12	~~4~~

$\frac{n}{2}$

~~48~~(3) indices searched

~~8~~ n

$\frac{2^n}{2}(n)$

8(4)

16

$\frac{2^4}{2}(4)$

8(4)

n=3 # sub seq = 8   n=4 # sub seq = 16   AND n=2 # sub seq = 4

So the patter he is 2^n of subsequences of a sequence.

AND searching through each index takes steps $T(n)=2^{n-1}\cdot n$

PSEUDOCODE FOR LONGEST ~~INCREASING~~ POWERSET generates (subsequences)

this is psuedo code is the

S = <subsets of a sequence> // to generate per last page takes about 2^n time

best = ∅ , is_best = TRUE //2

For i in S: // 2^n

  if(i.size() > best.size()): //1

    For(j=1; j< i.size(); j++): //n

      if(i[j-1] >= i[j]): // 1 + max(2,0) = 3

        is_best = FALSE //1

        break //1

  if (is_best == FALSE): //1 + max(2,0) = 3

    best = i //1

    is_best = TRUE //1

$$T(n) = 2 + 2^n \cdot 1 \cdot n \cdot 3 + 3 = 3 \cdot 2^n \cdot n + 5$$

To prove $T(n) = O(f(n))$ when $f(n) = 2^n \cdot n$

$$\lim_{n \to \infty} \frac{T'(n)}{f'(n)} = \frac{\frac{d}{dn}(3 \cdot 2^n \cdot n + 5)}{\frac{d}{dn}(2^n \cdot n)} \rightarrow \lim_{n \to \infty} \frac{3 \cdot 2^n \ln(2) \cdot n + 3 \cdot 2^n}{2^n \cdot \ln(2) \cdot n + 2^n} = \frac{3\cancel{(2^n \ln(2) \cdot n + 2^n)}}{\cancel{(2^n \cdot \ln(2) n + 2^n)}}$$

$$= 3$$

$$\therefore 3 \cdot 2^n \cdot n + 5 = O(2^n \cdot n)$$