Power Set Algorithm

```
powerSet (A[], R[])
3   [n = A.length, i = 0, max = 0;   1+1+1
1 - [ float [] temp = new float[n];
1 - temp[K]  = A[0];

for(j = 0; j < n; j++, i = 0) {
    for(K = 1; K < n; K++) {
        //add all values bigger than A[i] to temp
        if(temp[i] <= A[K])
            i++;
            temp[i] = A[K];
    }

    if(max < (i + 1)){
        System.arraycopy(temp, 0, R, n);
        max = (i+1);   1+1
    }
return max
```

Annotations:

$3n^2+3n$ (outer loop)

$3n-3$ ◄ $1+\max(3,0)$ (if block: 3)

$\sum_{K=1}^{n-1} 3 \Downarrow \sum_{K=\phi}^{n-1} 3$

$3*(n-1-1+1) = 3n-3$

$\sum_{j=0}^{n-1} 3n + \sum_{j=0}^{n-1} 3 = 3n(n) + 3(n)$

$n+4$: $\max(n+2)+2$; System.arraycopy $\rightarrow O(n)$

return max: 1

$3+1+1+3n^2+3n+n+4+1$

$PowerSet = 3n^2+4n+10$

Power Set Algorithm

1 [int max = 1

1 [float [] bestSubSq = new float[];

$3n^2+4n+11$ [max = powerSet(A, bestSubSq);
1 + $3n^2+4n+10$

Let max = m

// Copy bestSubSq to R
1 [R = new Float [max]
$m+1$ [for (int i = 0 ; i < max; i++)] $\sum_{i=0}^{m} 1 = (m-0+1)1$
1 R[i] = bestSubSq[i];]

$1+1+3n^2+4n+11+1+m+1$

$3n^2+4n+15+m$

$3n^2+4n+15+m \in O(n^2+m)$

Where n number of values in original input

Where m number of the longest non-decreasing subsequence in A[]

$$\lim_{\substack{n\to\infty \\ m\to\infty}} 3n^2+4n+15+m = \lim_{\substack{n\to\infty \\ m\to\infty}} \frac{3n^2+4n+15}{n^2} + \lim_{\substack{n\to\infty \\ m\to\infty}} \frac{m}{m}$$

= 3 + 1 > 0 and a constant

therefore $3n^2 + 4n + 15 \in O(n^2)$ and $m \in O(m)$
then $O(n^2) + O(m) = O(n^2 + m)$

Power set algorithm