

遞迴，問題越來越小，仍是同個問題，程式碼較精簡  
階乘、最大公因數、搜尋、費式數列、組合數  
河內塔

```
void writeBackward (string s, int size) ABC
{
    if (size > 0) {
        cout << s.substr (size-1, 1); 輸出最後一個字元 → C → B → A
        writeBackward (s, size-1); 遞迴呼叫
    } // end if
}
```

// size == 0 is the base case → do nothing

// end writeBackward

a > b 算 a 加到 b 的每個 a 總合

```
in sum (int a, int b) {
    if (a > b)
        return sum (a-1, b) + a;
    else → // a = b
        return a;
}
```

$$\begin{matrix} 100, 90 \\ \downarrow \\ 90, 90 \\ \downarrow \\ 80, 90 \\ \vdots \\ 90, 90 \end{matrix}$$

$$((90+91)+92)+\dots+(100)$$

```
int sum (int a, int b) {
    if (a == b)
        return a;
    return sum (a, b+1) + b;
}
```

從大往小加

求最大公因數

```
int gcd1 (int x, int y) {
    if (y == 0) return x;
    else if (y > x) return gcd1 (x, y % x);
    else return gcd1 (y, x % y);
}
```

```
int gcd2 (int x, int y) {
    if (! (x % y)) return y; // 整除
    else return gcd2 (y, x % y);
}
```

}

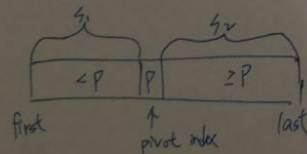
設 x = 9, y = 6  
 gcd1 (6, 3) →  
 x = 6, y = 3  
 gcd1 (3, 0) →  
 x = 3, y = 0  
 return 3

x = 9, y = 6  
 gcd2 (6, 3) →  
 x = 6, y = 3  
 return 3

效率：比次數

找第 k 小的數

selecting a pivot item in the array 再挑一個當 P



找第 k 小的數 又找其中一邊

if (k > s1) → 在右邊

if (k < s1) → 在左邊

7 25 39 19 48 2 16 12  
 30 7 25 19 2 16 12 || 48 39  
 左 右

12 7 25 19 2 16 30 48 39  
 4  
 12 7 25 19 16  
 2 7 12 25 19 16

```

kSmall(k, anArray, first, last)
= kSmall(k, anArray, first, pivotIndex - 1)
    if k < pivotIndex - first + 1 // 落在左半
= p    if k = pivotIndex - first + 1
= kSmall(k - (pivotIndex - first + 1), anArray, pivotIndex + 1, last)
    if k > pivotIndex - first + 1 // 落在右半

```

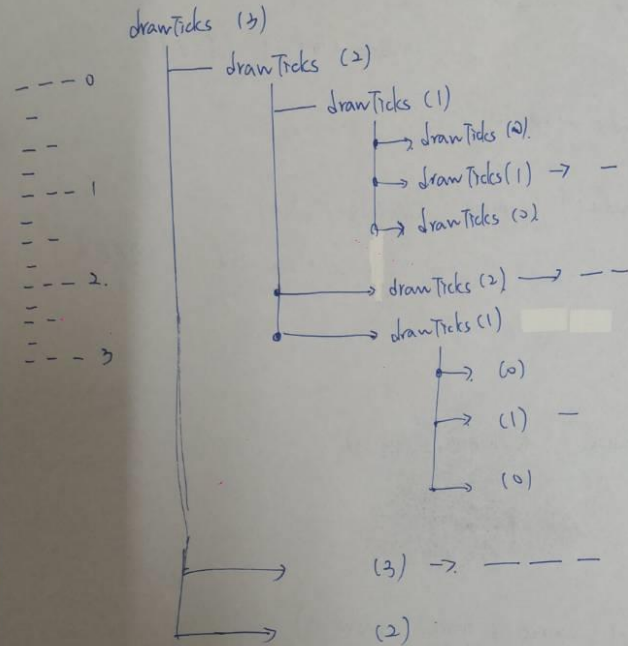
河内塔

```

solveTowers(count, source, destination, spare)
    if (count == 1)
        move a disk source to destination.
    else {
        solveTowers(count - 1, source, spare, destination)
        solveTowers(1, source, destination, spare)
        solveTowers(count - 1, spare, destination, source)
    }
}

```

Binary recursion: draw Ticks



递归  
 0.1 → 1.1  
 0.2 → 1.2  
 0.3 → 1.3  
 0.4 → 1.4  
 0.5 → 1.5  
 0.6 → 1.6  
 0.7 → 1.7  
 0.8 → 1.8  
 0.9 → 1.9  
 0.10 → 1.10  
 0.11 → 1.11  
 0.12 → 1.12  
 0.13 → 1.13  
 0.14 → 1.14  
 0.15 → 1.15  
 0.16 → 1.16  
 0.17 → 1.17  
 0.18 → 1.18  
 0.19 → 1.19  
 0.20 → 1.20  
 0.21 → 1.21  
 0.22 → 1.22  
 0.23 → 1.23  
 0.24 → 1.24  
 0.25 → 1.25  
 0.26 → 1.26  
 0.27 → 1.27  
 0.28 → 1.28  
 0.29 → 1.29  
 0.30 → 1.30  
 0.31 → 1.31  
 0.32 → 1.32  
 0.33 → 1.33  
 0.34 → 1.34  
 0.35 → 1.35  
 0.36 → 1.36  
 0.37 → 1.37  
 0.38 → 1.38  
 0.39 → 1.39  
 0.40 → 1.40  
 0.41 → 1.41  
 0.42 → 1.42  
 0.43 → 1.43  
 0.44 → 1.44  
 0.45 → 1.45  
 0.46 → 1.46  
 0.47 → 1.47  
 0.48 → 1.48  
 0.49 → 1.49  
 0.50 → 1.50  
 0.51 → 1.51  
 0.52 → 1.52  
 0.53 → 1.53  
 0.54 → 1.54  
 0.55 → 1.55  
 0.56 → 1.56  
 0.57 → 1.57  
 0.58 → 1.58  
 0.59 → 1.59  
 0.60 → 1.60  
 0.61 → 1.61  
 0.62 → 1.62  
 0.63 → 1.63  
 0.64 → 1.64  
 0.65 → 1.65  
 0.66 → 1.66  
 0.67 → 1.67  
 0.68 → 1.68  
 0.69 → 1.69  
 0.70 → 1.70  
 0.71 → 1.71  
 0.72 → 1.72  
 0.73 → 1.73  
 0.74 → 1.74  
 0.75 → 1.75  
 0.76 → 1.76  
 0.77 → 1.77  
 0.78 → 1.78  
 0.79 → 1.79  
 0.80 → 1.80  
 0.81 → 1.81  
 0.82 → 1.82  
 0.83 → 1.83  
 0.84 → 1.84  
 0.85 → 1.85  
 0.86 → 1.86  
 0.87 → 1.87  
 0.88 → 1.88  
 0.89 → 1.89  
 0.90 → 1.90  
 0.91 → 1.91  
 0.92 → 1.92  
 0.93 → 1.93  
 0.94 → 1.94  
 0.95 → 1.95  
 0.96 → 1.96  
 0.97 → 1.97  
 0.98 → 1.98  
 0.99 → 1.99  
 1.00 → 2.00  
 1.01 → 2.01  
 1.02 → 2.02  
 1.03 → 2.03  
 1.04 → 2.04  
 1.05 → 2.05  
 1.06 → 2.06  
 1.07 → 2.07  
 1.08 → 2.08  
 1.09 → 2.09  
 1.10 → 2.10  
 1.11 → 2.11  
 1.12 → 2.12  
 1.13 → 2.13  
 1.14 → 2.14  
 1.15 → 2.15  
 1.16 → 2.16  
 1.17 → 2.17  
 1.18 → 2.18  
 1.19 → 2.19  
 1.20 → 2.20  
 1.21 → 2.21  
 1.22 → 2.22  
 1.23 → 2.23  
 1.24 → 2.24  
 1.25 → 2.25  
 1.26 → 2.26  
 1.27 → 2.27  
 1.28 → 2.28  
 1.29 → 2.29  
 1.30 → 2.30  
 1.31 → 2.31  
 1.32 → 2.32  
 1.33 → 2.33  
 1.34 → 2.34  
 1.35 → 2.35  
 1.36 → 2.36  
 1.37 → 2.37  
 1.38 → 2.38  
 1.39 → 2.39  
 1.40 → 2.40  
 1.41 → 2.41  
 1.42 → 2.42  
 1.43 → 2.43  
 1.44 → 2.44  
 1.45 → 2.45  
 1.46 → 2.46  
 1.47 → 2.47  
 1.48 → 2.48  
 1.49 → 2.49  
 1.50 → 2.50  
 1.51 → 2.51  
 1.52 → 2.52  
 1.53 → 2.53  
 1.54 → 2.54  
 1.55 → 2.55  
 1.56 → 2.56  
 1.57 → 2.57  
 1.58 → 2.58  
 1.59 → 2.59  
 1.60 → 2.60  
 1.61 → 2.61  
 1.62 → 2.62  
 1.63 → 2.63  
 1.64 → 2.64  
 1.65 → 2.65  
 1.66 → 2.66  
 1.67 → 2.67  
 1.68 → 2.68  
 1.69 → 2.69  
 1.70 → 2.70  
 1.71 → 2.71  
 1.72 → 2.72  
 1.73 → 2.73  
 1.74 → 2.74  
 1.75 → 2.75  
 1.76 → 2.76  
 1.77 → 2.77  
 1.78 → 2.78  
 1.79 → 2.79  
 1.80 → 2.80  
 1.81 → 2.81  
 1.82 → 2.82  
 1.83 → 2.83  
 1.84 → 2.84  
 1.85 → 2.85  
 1.86 → 2.86  
 1.87 → 2.87  
 1.88 → 2.88  
 1.89 → 2.89  
 1.90 → 2.90  
 1.91 → 2.91  
 1.92 → 2.92  
 1.93 → 2.93  
 1.94 → 2.94  
 1.95 → 2.95  
 1.96 → 2.96  
 1.97 → 2.97  
 1.98 → 2.98  
 1.99 → 2.99  
 2.00 → 3.00  
 2.01 → 3.01  
 2.02 → 3.02  
 2.03 → 3.03  
 2.04 → 3.04  
 2.05 → 3.05  
 2.06 → 3.06  
 2.07 → 3.07  
 2.08 → 3.08  
 2.09 → 3.09  
 2.10 → 3.10  
 2.11 → 3.11  
 2.12 → 3.12  
 2.13 → 3.13  
 2.14 → 3.14  
 2.15 → 3.15  
 2.16 → 3.16  
 2.17 → 3.17  
 2.18 → 3.18  
 2.19 → 3.19  
 2.20 → 3.20  
 2.21 → 3.21  
 2.22 → 3.22  
 2.23 → 3.23  
 2.24 → 3.24  
 2.25 → 3.25  
 2.26 → 3.26  
 2.27 → 3.27  
 2.28 → 3.28  
 2.29 → 3.29  
 2.30 → 3.30  
 2.31 → 3.31  
 2.32 → 3.32  
 2.33 → 3.33  
 2.34 → 3.34  
 2.35 → 3.35  
 2.36 → 3.36  
 2.37 → 3.37  
 2.38 → 3.38  
 2.39 → 3.39  
 2.40 → 3.40  
 2.41 → 3.41  
 2.42 → 3.42  
 2.43 → 3.43  
 2.44 → 3.44  
 2.45 → 3.45  
 2.46 → 3.46  
 2.47 → 3.47  
 2.48 → 3.48  
 2.49 → 3.49  
 2.50 → 3.50  
 2.51 → 3.51  
 2.52 → 3.52  
 2.53 → 3.53  
 2.54 → 3.54  
 2.55 → 3.55  
 2.56 → 3.56  
 2.57 → 3.57  
 2.58 → 3.58  
 2.59 → 3.59  
 2.60 → 3.60  
 2.61 → 3.61  
 2.62 → 3.62  
 2.63 → 3.63  
 2.64 → 3.64  
 2.65 → 3.65  
 2.66 → 3.66  
 2.67 → 3.67  
 2.68 → 3.68  
 2.69 → 3.69  
 2.70 → 3.70  
 2.71 → 3.71  
 2.72 → 3.72  
 2.73 → 3.73  
 2.74 → 3.74  
 2.75 → 3.75  
 2.76 → 3.76  
 2.77 → 3.77  
 2.78 → 3.78  
 2.79 → 3.79  
 2.80 → 3.80  
 2.81 → 3.81  
 2.82 → 3.82  
 2.83 → 3.83  
 2.84 → 3.84  
 2.85 → 3.85  
 2.86 → 3.86  
 2.87 → 3.87  
 2.88 → 3.88  
 2.89 → 3.89  
 2.90 → 3.90  
 2.91 → 3.91  
 2.92 → 3.92  
 2.93 → 3.93  
 2.94 → 3.94  
 2.95 → 3.95  
 2.96 → 3.96  
 2.97 → 3.97  
 2.98 → 3.98  
 2.99 → 3.99  
 3.00 → 4.00  
 3.01 → 4.01  
 3.02 → 4.02  
 3.03 → 4.03  
 3.04 → 4.04  
 3.05 → 4.05  
 3.06 → 4.06  
 3.07 → 4.07  
 3.08 → 4.08  
 3.09 → 4.09  
 3.10 → 4.10  
 3.11 → 4.11  
 3.12 → 4.12  
 3.13 → 4.13  
 3.14 → 4.14  
 3.15 → 4.15  
 3.16 → 4.16  
 3.17 → 4.17  
 3.18 → 4.18  
 3.19 → 4.19  
 3.20 → 4.20  
 3.21 → 4.21  
 3.22 → 4.22  
 3.23 → 4.23  
 3.24 → 4.24  
 3.25 → 4.25  
 3.26 → 4.26  
 3.27 → 4.27  
 3.28 → 4.28  
 3.29 → 4.29  
 3.30 → 4.30  
 3.31 → 4.31  
 3.32 → 4.32  
 3.33 → 4.33  
 3.34 → 4.34  
 3.35 → 4.35  
 3.36 → 4.36  
 3.37 → 4.37  
 3.38 → 4.38  
 3.39 → 4.39  
 3.40 → 4.40  
 3.41 → 4.41  
 3.42 → 4.42  
 3.43 → 4.43  
 3.44 → 4.44  
 3.45 → 4.45  
 3.46 → 4.46  
 3.47 → 4.47  
 3.48 → 4.48  
 3.49 → 4.49  
 3.50 → 4.50  
 3.51 → 4.51  
 3.52 → 4.52  
 3.53 → 4.53  
 3.54 → 4.54  
 3.55 → 4.55  
 3.56 → 4.56  
 3.57 → 4.57  
 3.58 → 4.58  
 3.59 → 4.59  
 3.60 → 4.60  
 3.61 → 4.61  
 3.62 → 4.62  
 3.63 → 4.63  
 3.64 → 4.64  
 3.65 → 4.65  
 3.66 → 4.66  
 3.67 → 4.67  
 3.68 → 4.68  
 3.69 → 4.69  
 3.70 → 4.70  
 3.71 → 4.71  
 3.72 → 4.72  
 3.73 → 4.73  
 3.74 → 4.74  
 3.75 → 4.75  
 3.76 → 4.76  
 3.77 → 4.77  
 3.78 → 4.78  
 3.79 → 4.79  
 3.80 → 4.80  
 3.81 → 4.81  
 3.82 → 4.82  
 3.83 → 4.83  
 3.84 → 4.84  
 3.85 → 4.85  
 3.86 → 4.86  
 3.87 → 4.87  
 3.88 → 4.88  
 3.89 → 4.89  
 3.90 → 4.90  
 3.91 → 4.91  
 3.92 → 4.92  
 3.93 → 4.93  
 3.94 → 4.94  
 3.95 → 4.95  
 3.96 → 4.96  
 3.97 → 4.97  
 3.98 → 4.98  
 3.99 → 4.99  
 4.00 → 5.00  
 4.01 → 5.01  
 4.02 → 5.02  
 4.03 → 5.03  
 4.04 → 5.04  
 4.05 → 5.05  
 4.06 → 5.06  
 4.07 → 5.07  
 4.08 → 5.08  
 4.09 → 5.09  
 4.10 → 5.10  
 4.11 → 5.11  
 4.12 → 5.12  
 4.13 → 5.13  
 4.14 → 5.14  
 4.15 → 5.15  
 4.16 → 5.16  
 4.17 → 5.17  
 4.18 → 5.18  
 4.19 → 5.19  
 4.20 → 5.20  
 4.21 → 5.21  
 4.22 → 5.22  
 4.23 → 5.23  
 4.24 → 5.24  
 4.25 → 5.25  
 4.26 → 5.26  
 4.27 → 5.27  
 4.28 → 5.28  
 4.29 → 5.29  
 4.30 → 5.30  
 4.31 → 5.31  
 4.32 → 5.32  
 4.33 → 5.33  
 4.34 → 5.34  
 4.35 → 5.35  
 4.36 → 5.36  
 4.37 → 5.37  
 4.38 → 5.38  
 4.39 → 5.39  
 4.40 → 5.40  
 4.41 → 5.41  
 4.42 → 5.42  
 4.43 → 5.43  
 4.44 → 5.44  
 4.45 → 5.45  
 4.46 → 5.46  
 4.47 → 5.47  
 4.48 → 5.48  
 4.49 → 5.49  
 4.50 → 5.50  
 4.51 → 5.51  
 4.52 → 5.52  
 4.53 → 5.53  
 4.54 → 5.54  
 4.55 → 5.55  
 4.56 → 5.56  
 4.57 → 5.57  
 4.58 → 5.58  
 4.59 → 5.59  
 4.60 → 5.60  
 4.61 → 5.61  
 4.62 → 5.62  
 4.63 → 5.63  
 4.64 → 5.64  
 4.65 → 5.65  
 4.66 → 5.66  
 4.67 → 5.67  
 4.68 → 5.68  
 4.69 → 5.69  
 4.70 → 5.70  
 4.71 → 5.71  
 4.72 → 5.72  
 4.73 → 5.73  
 4.74 → 5.74  
 4.75 → 5.75  
 4.76 → 5.76  
 4.77 → 5.77  
 4.78 → 5.78  
 4.79 → 5.79  
 4.80 → 5.80  
 4.81 → 5.81  
 4.82 → 5.82  
 4.83 → 5.83  
 4.84 → 5.84  
 4.85 → 5.85  
 4.86 → 5.86  
 4.87 → 5.87  
 4.88 → 5.88  
 4.89 → 5.89  
 4.90 → 5.90  
 4.91 → 5.91  
 4.92 → 5.92  
 4.93 → 5.93  
 4.94 → 5.94  
 4.95 → 5.95  
 4.96 → 5.96  
 4.97 → 5.97  
 4.98 → 5.98  
 4.99 → 5.99  
 5.00 → 6.00  
 5.01 → 6.01  
 5.02 → 6.02  
 5.03 → 6.03  
 5.04 → 6.04  
 5.05 → 6.05  
 5.06 → 6.06  
 5.07 → 6.07  
 5.08 → 6.08  
 5.09 → 6.09  
 5.10 → 6.10  
 5.11 → 6.11  
 5.12 → 6.12  
 5.13 → 6.13  
 5.14 → 6.14  
 5.15 → 6.15  
 5.16 → 6.16  
 5.17 → 6.17  
 5.18 → 6.18  
 5.19 → 6.19  
 5.20 → 6.20  
 5.21 → 6.21  
 5.22 → 6.22  
 5.23 → 6.23  
 5.24 → 6.24  
 5.25 → 6.25  
 5.26 → 6.26  
 5.27 → 6.27  
 5.28 → 6.28  
 5.29 → 6.29  
 5.30 → 6.30  
 5.31 → 6.31  
 5.32 → 6.32  
 5.33 → 6.33  
 5.34 → 6.34  
 5.35 → 6.35  
 5.36 → 6.36  
 5.37 → 6.37  
 5.38 → 6.38  
 5.39 → 6.39  
 5.40 → 6.40  
 5.41 → 6.41  
 5.42 → 6.42  
 5.43 → 6.43  
 5.44 → 6.44  
 5.45 → 6.45  
 5.46 → 6.46  
 5.47 → 6.47  
 5.48 → 6.48  
 5.49 → 6.49  
 5.50 → 6.50  
 5.51 → 6.51  
 5.52 → 6.52  
 5.53 → 6.53  
 5.54 → 6.54  
 5.55 → 6.55  
 5.56 → 6.56  
 5.57 → 6.57  
 5.58 → 6.58  
 5.59 → 6.59  
 5.60 → 6.60  
 5.61 → 6.61  
 5.62 → 6.62  
 5.63 → 6.63  
 5.64 → 6.64  
 5.65 → 6.65  
 5.66 → 6.66  
 5.67 → 6.67  
 5.68 → 6.68  
 5.69 → 6.69  
 5.70 → 6.70  
 5.71 → 6.71  
 5.72 → 6.72  
 5.73 → 6.73  
 5.74 → 6.74  
 5.75 → 6.75  
 5.76 → 6.76  
 5.77 → 6.77  
 5.78 → 6.78  
 5.79 → 6.79  
 5.80 → 6.80  
 5.81 → 6.81  
 5.82 → 6.82  
 5.83 → 6.83  
 5.84 → 6.84  
 5.85 → 6.85  
 5.86 → 6.86  
 5.87 → 6.87  
 5.88 → 6.88  
 5.89 → 6.89  
 5.90 → 6.90  
 5.91 → 6.91  
 5.92 → 6.92  
 5.93 → 6.93  
 5.94 → 6.94  
 5.95 → 6.95  
 5.96 → 6.96  
 5.97 → 6.97  
 5.98 → 6.98  
 5.99 → 6.99  
 6.00 → 7.00  
 6.01 → 7.01  
 6.02 → 7.02  
 6.03 → 7.03  
 6.04 → 7.04  
 6.05 → 7.05  
 6.06 → 7.06  
 6.07 → 7.07  
 6.08 → 7.08  
 6.09 → 7.09  
 6.10 → 7.10  
 6.11 → 7.11  
 6.12 → 7.12  
 6.13 → 7.13  
 6.14 → 7.14  
 6.15 → 7.15  
 6.16 → 7.16  
 6.17 → 7.17  
 6.18 → 7.18  
 6.19 → 7.19  
 6.20 → 7.20  
 6.21 → 7.21  
 6.22 → 7.22  
 6.23 → 7.23  
 6.24 → 7.24  
 6.25 → 7.25  
 6.26 → 7.26  
 6.27 → 7.27  
 6.28 → 7.28  
 6.29 → 7.29  
 6.30 → 7.30  
 6.31 → 7.31  
 6.32 → 7.32  
 6.33 → 7.33  
 6.34 → 7.34  
 6.35 → 7.35  
 6.36 → 7.36  
 6.37 → 7.37  
 6.38 → 7.38  
 6.39 → 7.39  
 6.40 → 7.40  
 6.41 → 7.41  
 6.42 → 7.42  
 6.43 → 7.43  
 6.44 → 7.44  
 6.45 → 7.45  
 6.46 → 7.46  
 6.47 → 7.47  
 6.48 → 7.48  
 6.49 → 7.49  
 6.50 → 7.50  
 6.51 → 7.51  
 6.52 → 7.52  
 6.53 → 7.53  
 6.54 → 7.54  
 6.55 → 7.55  
 6.56 → 7.56  
 6.57 → 7.57  
 6.58 → 7.58  
 6.59 → 7.59  
 6.60 → 7.60  
 6.61 → 7.61  
 6.62 → 7.62  
 6.63 → 7.63  
 6.64 → 7.64  
 6.65 → 7.65  
 6.66 → 7.66  
 6.67 → 7.67  
 6.68 → 7.68  
 6.69 → 7.69  
 6.70 → 7.70  
 6.71 → 7.71  
 6.72 → 7.72  
 6.73 → 7.73  
 6.74 → 7.74  
 6.75 → 7.75  
 6.76 → 7.76  
 6.77 → 7.77  
 6.78 → 7.78  
 6.79 → 7.79  
 6.80 → 7.80  
 6.81 → 7.81  
 6.82 → 7.82  
 6.83 → 7.83  
 6.84 → 7.84  
 6.85 → 7.85  
 6.86 → 7.86  
 6.87 → 7.87  
 6.88 → 7.88  
 6.89 → 7.89  
 6.90 → 7.90  
 6.91 → 7.91  
 6.92 → 7.92  
 6.93 → 7.93  
 6.94 → 7.94  
 6.95 → 7.95  
 6.96 → 7.96  
 6.97 → 7.97  
 6.98 → 7.98  
 6.99 → 7.99  
 7.00 → 8.00  
 7.01 → 8.01  
 7.02 → 8.02  
 7.03 → 8.03  
 7.04 → 8.04  
 7.05 → 8.05  
 7.06 → 8.06  
 7.07 → 8.07  
 7.08 → 8.08  
 7.09 → 8.09  
 7.10 → 8.10  
 7.11 → 8.11  
 7.12 → 8.12  
 7.13 → 8.13  
 7.14 → 8.14  
 7.15 → 8.15  
 7.16 → 8.16  
 7.17 → 8.17  
 7.18 → 8.18  
 7.19 → 8.19  
 7.20 → 8.20  
 7.21 → 8.21  
 7.22 → 8.22  
 7.23 → 8.23  
 7.24 → 8.24  
 7.25 → 8.25  
 7.26 → 8.26  
 7.27 → 8.27  
 7.28 → 8.28  
 7.29 → 8.29  
 7.30 → 8.30  
 7.31 → 8.31  
 7.32 → 8.32  
 7.33 → 8.33  
 7.34 → 8.34  
 7.35 → 8.35  
 7.36 → 8.36  
 7.37 → 8.37  
 7.38 → 8.38  
 7.39 → 8.39  
 7.40 → 8.40  
 7.41 → 8.41  
 7.42 → 8.42  
 7.43 → 8.43  
 7.44 → 8.44  
 7.45 → 8.45  
 7.46 → 8.46  
 7.47 → 8.47  
 7.48 → 8.48  
 7.49 → 8.49  
 7.50 → 8.50  
 7.51 → 8.51  
 7.52 → 8.52  
 7.53 → 8.53  
 7.54 → 8.54  
 7.55 → 8.55  
 7.56 → 8.56  
 7.57 → 8.57  
 7.58 → 8.58  
 7.59 → 8.59  
 7.60 → 8.60  
 7.61 → 8.61  
 7.62 → 8.62  
 7.63 → 8.63  
 7.64 → 8.64  
 7.65 → 8.65  
 7.66 → 8.66  
 7.67 → 8.67  
 7.68 → 8.68  
 7.69 → 8.69  
 7.70 → 8.70  
 7.71 → 8.71  
 7.72 → 8.72  
 7.73 → 8.73  
 7.74 → 8.74  
 7.75 → 8.75  
 7.76 → 8.76  
 7.77 → 8.77  
 7.78 → 8.78  
 7.79 → 8.79  
 7.80 → 8.80  
 7.81 → 8.81  
 7.82 → 8.82  
 7.83 → 8.83  
 7.84 → 8.84  
 7.85 → 8.85  
 7.86 → 8.86  
 7.87 → 8.87  
 7.88 → 8.88  
 7.89 → 8.89  
 7.90 → 8.90  
 7.91 →



# 費式數列

$$n_1 = 1$$

$$n_2 = 1$$

$$n_3 = n_2 + n_1 + 1 = 3$$

$$n_4 = n_3 + n_2 + 1 = 5$$

$$n_5 = n_4 + n_3 + 1 = 9$$

:

$n_k$  at least doubles every other time

$n_k > 2^{\frac{k}{2}}$ , exponential 呼叫次數以指數成長

## 求指數函數

(a) 以迴圈求  $n$  次方

(b) 以遞迴:  $x^0 = 1$ ,  $x^n = x * x^{n-1}$ , if  $n > 0$

(c) 二元遞迴  $x^0 = 1$ ,  $x^n = (x^{\frac{n}{2}})^2$ , if  $n > 0$  and  $n$  偶數  
 $x^n = x * (x^{\frac{n-1}{2}})^2$ , if  $n > 0$  and  $n$  奇數

(a) double power1 (double x, int n) {  
 double result = 1;  
 while (n > 0) {  
 result = result \* x;  
 n --;  
 }  
 return result;  
}

$$9^4 = (((1 * 9) * 9) * 9) * 9$$

(b) double power2 (double x, int n) {  
 if (n == 0)  
 return 1;  
 else  
 return x \* power2 (x, n-1);  
}

$$9^4 = 9 * (9 * (9 * (9 * 1)))$$

4 乘法次數  
5 次呼叫

(c) double power3 (double x, int n) {

if (n == 0)  
 return 1;

→ if (n == 1)  
 return x;

else {

double halfpower = power3 (x, n/2);

if (n % 2 == 0) // 偶  
 return halfpower \* halfpower;

else // 奇

return x \* halfpower \* halfpower;

}

}

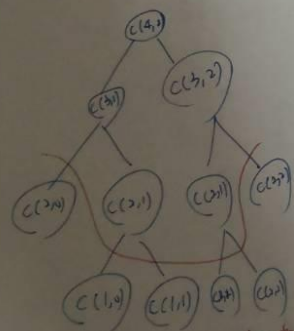
$$9^4 = ((9 * 1 * 1) * 9) * 9$$

4 乘法次數  
4 次呼叫

$$9^4 = (9 * 9) * 9 * 9$$

2 乘法次數  
3 次呼叫

n 選 k



遞迴呼叫:

$$2 * C(n, k) - 1$$

$$|leaf\ nodes| - |internal\ node| = 1$$

相消後剩餘的為樹葉

所有東西都是物件

adj or n. : data members  
v or v-ing : methods

classes of objects:  
Attributes = data members  
Behaviors = methods

Encapsulation: hides inner details

Inheritance: reused (概念: 超連結)

ADT List

createList () 建構

destroyList () 解構

isEmpty () : boolean {query} 是否為空

getLength () : integer {query} 計算長度

insert (in index: integer, in newItem: ListItemType,  
out success: boolean) 插入

remove (in index: integer, out success: boolean) 刪除

retrieve (in index: integer, out dataItem: ListItemType,  
out success: boolean) {query} 檢索

Reverse the entire list

reverseList (in aList: List, out source: boolean)  
for cm=1 to aList.getLength()-1

{ aList.retrieve (1, dataItem, success);  
aList.remove (1, success);  
aList.insert (aList.getLength()-it+1,  
dataItem, success);

}

reverseList2 (in aList: List, out source: boolean)

for (i=1 to aList.getLength()-1)

{ aList.retrieve (aList.getLength(), dataItem, success);  
aList.insert (\_\_\_\_, dataItem, success);  
aList.remove (aList.getLength()+1, success);

}

degree () : integer

{ return aList.getLength()-1;  
}

coefficient (in power: integer) : integer

{ aList.retrieve (power+1, aCoefficient, success);  
if (success) return aCoefficient;  
else return 0;

}



Delete a Node from a Sorted Linked list

```
Node *prev, *cur;
if (head != NULL)
{
    for (prev = NULL, cur = head;
        (cur != NULL) && (newVal > cur->item);
        prev = cur, cur = cur->next;
    )
    {
        if (prev == NULL)
            head = cur->next;
        else prev->next = cur->next;
        cur->next = NULL;
        delete cur;
    }
}
```

已排列  
走訪至節點

改變指標

歸還空間

ofstream outFile(fileName); 宣告輸出檔案

```
for (Node *cur = head; cur != NULL; cur = cur->next)
    outFile << cur->item << endl;
outFile.close();
```

逐行寫檔, 存資料

關閉輸出檔案

```
ifstream inFile(fileName);
int nextItem;
if (inFile >> nextItem)
{
    head->item = nextItem;
    head->next = NULL;
    tail = head;
}
try
```

head->item = nextItem;

head->next = NULL;

tail = head; 第一個節點

Infix expressions

ex: a + b

中序運算式

Prefix expressions

ex: + a b

前序

Postfix expressions

ex: a b +

後序

bool Map::isPath(int originCity, int destinationCity)

```
{
    int nextCity;
    bool success, done;
```

markVisited(originCity);

if (originCity == destinationCity) 抵達目的地

return true;

else

```
{
    done = false;
```

success = getNearestCity(originCity, nextCity); 無航班  
重複的城市