Question 1.

1. 最大子段和问题：给定整数序列 $a_1, a_2, \cdots, a_n$，求该序列形如 $\sum_{k=i}^{j} a_k$ 的子段和的最大值：$\max\left\{0, \max_{1 \le i \le j \le n} \sum_{k=i}^{j} a_k\right\}$

1）已知一个简单算法如下：

```
int Maxsum(int n, int a, int& besti, int& bestj)
{
   int sum = 0;
   for(int i=1;i<=n;i++){
     int suma = 0;
     for(int j=i;j<=n;j++){
        suma + = a[j];
        if(suma > sum){
          sum = suma;
          besti = i;
          bestj = j;
          }
        }
      }
     return sum;
}
```

试分析该算法的时间复杂性。

2）试用分治算法解最大子段和问题，并分析算法的时间复杂性。

3) 试说明最大子段和问题具有最优子结构性质，并设计一个动态规划算法解最大子段和问题。分析算法的时间复杂度。（提示：令 $b(j) = \max_{1 \le i \le j \le n} \sum_{k=i}^{j} a_k$，$j = 1, 2, \cdots, n$）。

My Answer 1. *(1)* 由题可知道算法为两重循环，特征步数为 $\frac{n(n-1)}{2}$，故时间复杂度为:$O(n^2)$ *(2)* 由题编写 *python* 代码，主要思路为将求解数组 $a(n)$ 的最大子段问题拆分为求解 $a(0:\frac{n}{2})$ 与 $a(\frac{n}{2}:n)$ 两个子段的子段问题。

针对每一层，最大子段只会有三种情况：完全包含于左子段；完全包含于右子段；包含左子段与右子段的分界线，因此只需要在归并的时候计算中间部分的子段，然后与其余两种情况作比较即可，实现效果如下：

```
(py36) D:\子钰\文案\个人\研究生\研一\计算机算法设计与分析\src>python fenzhi.py
给定数组为: { [5, 31, -35, 100, -5, -2, 10, -3, 1, -5, 6, -6, -10] }
最大子段和为 104
最大子段为第 0 到第 6 个元素
```

计算时间 $T(n)$ 满足分治算法的递归式：$T(n) = 2T(\frac{n}{2})+O(n)$，因此该算法的时间复杂度为 $O(n\log(n))$，代码说明：直接运行附件中的 *A11.py* 即可 *(3)* 设 $b(j) = \max_{1\le i\le j}\{\sum_{k=i}^{j} a_k\}$，则问题 $b(n)$ 与问题 $b(n-1)$ 具有关系：

$$
\begin{aligned}
b(j) &= \max_{1\le i\le j}\{\sum_{k=i}^{j} a_k\} \\
&= \max\{\{\sum_{k=i}^{j-1} a_k + a_j\}, a_j\}(1 \le i \le j-1) \\
&= \max\{b(j-1) + a_j, a_j\}
\end{aligned}
$$

因此具有最优子结构性质，动态规划的迭代公式为：$b(j) = \max\{b(j-1)+a_j, a_j\} 1 \le j \le n$，编写 *python* 代码，实现效果如下：

```
给定数组为: { [5, 31, -35, 100, -5, -2, 10, -3, 1, -5, 6, -6, -10] }
最大子段和为 104.0
```

代码说明：直接运行附件中的 *A13.py* 即可

Question 2.

2.（双机调度问题）用两台处理机 A 和 B 处理 n 个作业。设第 i 个作业交给机器 A 处理时所需要的时间是 a_i，若由机器 B 来处理，则所需要的时间是 b_i。

现在要求每个作业只能由一台机器处理，每台机器都不能同时处理两个作业。设计一个动态规划算法，使得这两台机器处理完这 n 个作业的时间最短（从任何一台机器开工到最后一台机器停工的总的时间）。以下面的例子说明你的算法：

$n = 6, (a_1, a_2, a_3, a_4, a_5, a_6) = (2, 5, 7, 10, 5, 2), (b_1, ..., b_6) = (3, 8, 4, 11, 3, 4)$

My Answer 2. 设 $T(i)$ 为完成前 i 个作业，所需要的最短处理时间；$T_a(i)$ 为在最优策略下，分配到机器 A 上 A 处理所需要的时间；$T_b(i)$ 为在最优策略下，分配到机器 B 上 B 处理所需要的时间；则 $T(i) = \max\{T_a(i), T_b(i)\}$，同时得到递推公式：$T(i) = \max\{T(i-1), \min\{T_a(i-1) + a_i, T_b(i-1) + b_i\}\}$ 则编写代码，实现效果如下：

```
a为:  [2, 5, 7, 10, 5, 2]
b为:  [3, 8, 4, 11, 3, 4]
最短时间为:  15
```

并实例说明：

$$i = 1, T_a(i-1) = 2, T_b(i-1) = 0 \Rightarrow T = 2$$

$$i = 2, T_a(i-1) = 7, T_b(i-1) = 0 \Rightarrow T = 7$$

$$i = 3, T_a(i-1) = 7, T_b(i-1) = 4 \Rightarrow T = 7$$

$$i = 4, T_a(i-1) = 7, T_b(i-1) = 15 \Rightarrow T = 15$$

$$i = 5, T_a(i-1) = 12, T_b(i-1) = 15 \Rightarrow T = 15$$

$$i = 6, T_a(i-1) = 14, T_b(i-1) = 15 \Rightarrow T = 15$$

因此最短时间为 *15*。代码说明：直接运行附件中的 *A2.py* 即可，可以看见代码中只有一重循环，因此复杂度为 $O(n)$

Question 3.

3．考虑下面特殊的整数线性规划问题

$$\max \sum_{i=1}^{n} c_i x_i$$

$$\sum_{i=1}^{n} a_i x_i \le b, \quad x_i \in \{0,1,2\}, 1 \le i \le n$$

试设计一个解此问题的动态规划算法，并分析算法的时间复杂度。

My Answer 3. 设 $y_i \in \{0,1\}, 1 \le i \le 2n$，令 $x_i = y_i + y_{i+n}, 1 \le i \le n$，则上述问题可以转化为：

$\max \sum_{i=1}^{2n} c_i y_i, S.t. \sum_{i=1}^{2n} a_i y_i \le b$

且有 $c_{i+n} = c_i, a_{i+n} = a_i$ $1 \le i \le n$，若将 c_i 看作价值，a_i 看作重量,b 看作容量，则该问题即为 *0/1* 背包问题。又由于 *0/1* 背包问题的动态规划算法复杂度为 $O(2^n)$，因此该算法复杂度为 $O(4^n)$

Question 4.

4．可靠性设计：一个系统由 n 级设备串联而成，为了增强可靠性，每级都可能并联了不止一台同样的设备。假设第 i 级设备 D_i 用了 m_i 台，该级设备的可靠性是 $g_i(m_i)$，则这个系统的可靠性是 $\Pi g_i(m_i)$。一般来说 $g_i(m_i)$都是递增函数，所以每级用的设备越多系统的可靠性越高。但是设备都是有成本的，假定设备 D_i 的成本是 c_i，设计该系统允许的投资不超过 c，那么，该如何设计该系统（即各级采用多少设备）使得这个系统的可靠性最高。试设计一个动态规划算法求解可靠性设计问题。

My Answer 4. 由题可得问题的约束条件：$\sum_{i=1}^{n} m_i c_i \leq c, 1 \leq m_i \leq 1 + [\frac{c-\sum_{i=1}^{n} c_i}{c_i}]$(也即每一级都至少要有一个设备)，以及优化目标：$\max \mathit{\Pi}_{i=1}^{n} g_i(m_i)$

记 $G[k](x)$ 为第 k 级设备在可用投资为 x 的系统可靠性最大值，则有：

$G[k](x) = \max_{1 \leq m_k \leq [\frac{x}{c_k}]} \{g_k(m_k) G[k-1](x - c_k m_k)\}$

同时定义 $G[0](x) = \begin{cases} 1, & \sum_{i=1}^{n} c_i \leq x \leq c \\ -\infty, & else \end{cases}$

则：$G[k](x) = \begin{cases} -\infty, & x < \sum_{i=k}^{n} c_i \\ \max\{g_j(m_k) G[k-1](x - c_k m_k)\}, & x \geq \sum_{i=k}^{n} c_i \end{cases}$

则依次求解动态规划算法，即可得到最终解。