第十章 贪心算法

1. 基本思想

贪心算法又称贪婪算法，是一种在每一步选择中都采取在当前状态下最好或最优 （即最有利） 的选择，从而期望得到的结果是最好或最优的算法。

2. 贪心算法适用的场景

贪心算法通常适用于满足***贪心选择性质***和***最优子结构性质***的问题
贪心算法没有固定的算法框架，关键在于贪心策略的选择，所采用的贪心策略要满足 ***无后效性***

3. 贪心算法的基本要素

最优子结构性质： 问题的最优解包含子问题的最优解。 这意味着，问题可以分解成若个子问题 每个子问题可以独立地求解，并且它们的最优解可以组合成原问题的最优解

贪心选择性质：通过每个子问题的最优选择，可以得到整个问题的最优解。这意味着，当我们面对一个问题时，我们可以通过贪心策略来做出局部最优的选择，最终得到全局最优的解

无后效性： 当我们做出一个选择后， 它对后面的选择没有影响。 这意味着，我们在做出一个选择 时，只需要考虑当前的局部最优解， 而不需要考虑将来的影响

4. 具体步骤：

1. 把求解的问题分成若个子问题；
2. 对每个子问题求解，得到子问题的局部最优解；
3. 把子问题的解局部最优解合成原问题的一个解

5. 常见应用

1. 零钱找零问题：给定不同面额的硬币和一个金额，找到成该金额所需的最少硬币数；
2. 背包问题：给定一组物品和一个背包的容量，选择一些物品放入背包中，使得总价值最大
3. 区间调度问题：给定一组区间，选择尽可能多的区间，使得它们互不重叠；
4. 哈夫曼编码：用变长编码表示字符，便得出现频率高的字符编码短， 出现频率低的学符编码长；
5. 最小生成树：在n个城市之间铺设光缆，要使这n个城市的任意两个之间都可以通信，目标是要使铺设光缆的总费用最低， 这就需要我到带权的最小生成树

6. 贪心算法总结

1. 贪心算法通常适用于求解最优化问题，但不是所有最优化问题都适用心算法；
2. 问题必须具备贪心选择性质和最优子结构性质，才能使用心算法求解
3. 贪心算法有时候不能得到全局最优解，但是贪心算法是一种简单而有效的算法思想，在某些问题中，贪心算法可以得到近似最优解；例如：人工智能很多算法都是心算法
4. 贪心算法实现比较容易，但是证明正确性很难。

7.示例：

小明有N元 （有零有整） 的纸币，想去银行将纸币兑换成硬币，银行硬币只有1角，5角，1元三种。请计算出最少换多少个硬币?
解题思路:
银行只提供了1角、5角和1元的硬币。根据心算法的性质，我们应该尽量使用面额最大的硬市来兑换 使兑换的硬币的数量更少

1）将纸币金额乘以 10，转换为整数，方便处理；
2）先尽可能多地兑换 1元硬币，即将纸币金额除以10，得到需要多少个1元硬，并更新剩余金额
3）再尽可能多地兑换 5角硬币，即将剩余金额除以5，得到需要多少个5角硬币，并更新剩余金额；
4）剩余金额即为需要的 1角硬币数量；
5）将三种硬币数量累加。
其实，示例就是将一个大问题分成若干个小问题，然后对每个小问题进行求解，最后将子问题的解合并，成 为原问题的解

参考代码：

```cpp
#include <iostream>
using namespace std;

int minimumCoins(double N) {
    int count = 0;              // 记录兑换的硬币数量
    int amount = N * 10;        // 将金额乘以10倍转换为整数

    int oneYuan = amount / 10; // 计算需要1元硬币数量
    count += oneYuan;
    amount -= oneYuan * 10;     // 更新剩余金额

    int fiveJiao = amount / 5; // 计算需要5角硬币数量
    count += fiveJiao;
    amount -= fiveJiao * 5;     // 更新剩余金额

    cout << "换取1元硬币" << oneYuan << "个" << endl;
    cout << "换取5角硬币" << fiveJiao << "个" << endl;
    cout << "换取1角硬币" << amount << "个" << endl;

    count += amount;          // 剩余金额即为需要的1角硬币数量

    return count;
}
```

```cpp
int main() {
    double N;
    cout << "请输入纸币金额：";
    cin >> N;
    //调用函数计算换取硬币数量
    int result = minimumCoins(N);
    cout << "最少换取硬币数为：" << result;

    return 0;
}
```

```
请输入纸币金额：2.7
换取1元硬币2个
换取5角硬币1个
换取1角硬币2个
最少换取硬币数为：5
```