# 深度优先遍历：

## 1、用处：

穷举出所有可能性，找出结果。

方案有许多时，并且求最优方案时，可以递归算出所有方案，求最优

## 2、用法：

每一个元素都可能选与不选，当前元素选作为一个递归入口，不选也作为一个递归入口。看哪个可行。

# 宽度优先遍历：

## 1、用处：

按照开始状态由近到远的距离开始搜索，可以很容易的来求最少操作，最短路径之类问题的答案。

## 2、用法：

使用队列实现。

# 贪心算法：

# 四、动态规划（DP）：

## 1、用处：

最优解问题，各种排列组合的个数，期望，概率的计算

## 2、用法：

1. m表（记录表的设计）：

用一个数组记录下算法中曾今计算过的结果，避免重复计算。

1. 递归式的推算：

## 3、记忆动态规划算法：

（1）m表可以是多维。

（2）几个下标表示当前的一个状态。

（3）参考2014-09.cpp 2014年C第9题

# 单源最短路径：

## Dijkstra算法：

选取距离源点s最近的一点v，加入已经确定最短的顶点集合中，刷新v的相邻顶点。

选取距离源点s最近的一点v，该顶点就是最短距离的？

因为顶点间距离不是负数，并且是能够到达s的边, 之后的更新中不会变小，因此，现在距离s最小的，就是最短路径了

# 回溯法：

1. 回溯法中，每一个节点可以重用，不能在当前回溯过程中计算出某一个值然后传给下一层递归。
2. 深度优先遍历，节点不重复遍历，可以在当前回溯过程中计算出某一个值然后传给下一层递归。