**基本要素**

分治法：最优子结构，...

贪心策略：贪心选择性质，最优子结构

动态规划：最优化原理(最优子结构)，重叠子问题，备忘录方法

回溯法：

最优子结构：某个问题的最优解包含了其子问题的最优解；

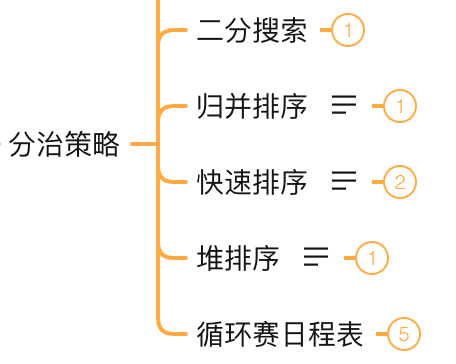
最优化原理：不管前面的决策是怎样的，以后的决策只取决于前面决策所决定的状态；

贪心选择性质：问题的整体解可以通过一系列局部最优的选择，即贪心选择达到；

重叠子问题：每次产生的子问题并不总是新问题，有些子问题被反复计算多次

**四大算法基本思想**

**分治法：**

是将一个规模为n的问题分解为k个规模较小的子问题，这些子问题互相独立且与原问题相同；对这k个子问题分别求解。如果子问题的规模仍然不够小，则再划分为k个子问题，如此递归的进行下去，直到问题规模足够小，很容易求出其解为止；将求出的小规模的问题的解合并为一个更大规模的问题的解，自底向上逐步求出原来问题的解。

**贪心策略：**

从问题的某一个初始解出发逐步逼近给定的目标，以尽可能快的求得更好的解。当达到某个算法中的某一步不能再继续前进时，算法停止。

**普里姆算法**

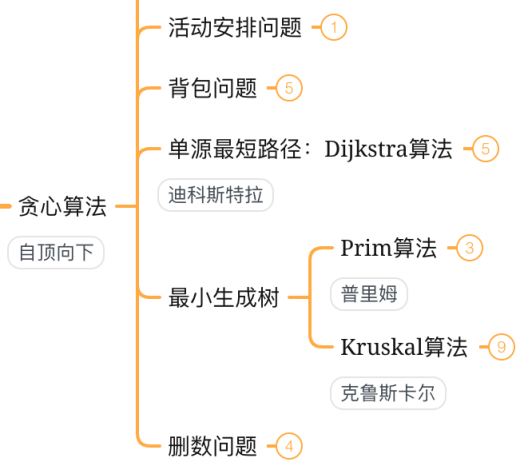
贪心策略：每次都选i∈U，j∈V-U，且c[i][j]最小的边

基本思想：设G=(V,E)是连通带权图，V={1,2,…,n}。首先置U={1}，然后，只要U是V的真子集，就选取满足条件i∈U，j∈V-U，且c[i][j]最小的边，将顶点j添加到U中。这个过程一直进行到U=V时为止。

**克鲁斯卡尔算法**

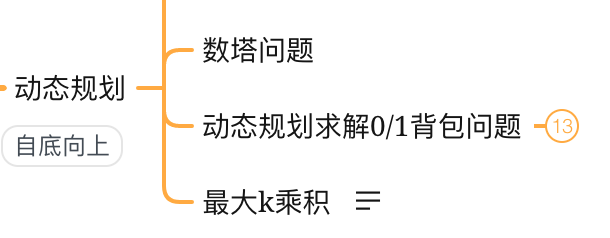
贪心策略：每次都在连接两个不同连通分量的边中选权值最小的边。

基本思想：首先将图中所有顶点都放到生成树中，然后每次都连接两个不同连通分量的边中选权值最小的边，将其放入生成树中，直到生成树中有n-1条边



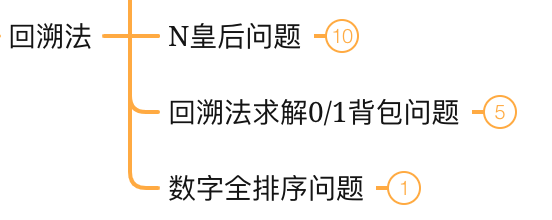
**动态规划：**

在求解问题中，对于每一步决策，列出各种可能的局部解，再依据某种判定条件，舍弃那些肯定不能得到最优解的局部解，在每一步都经过筛选，以每一步都是最优解来保证全局是最优解。动态规划是一种将问题实例分解为更小的、相似的子问题，动态规划允许这些子问题不独立，(亦即各子问题可包含公共的子问题)也允许其通过自身子问题的解做出选择，该方法对每一个子问题只解一次，并将结果保存起来，避免每次碰到时都要重复计算。



**回溯法：**

是在一棵含有问题全部可能解的状态空间树上进行深度优先搜索，解为叶子结点。

搜索过程中，每到达一个结点时，则判断该结点为根的子树是否含有问题的解，如果可以确定该子树中不含有问题的解，则放弃对该子树的搜索，退回到上层父结点，继续下一步深度优先搜索过程。在回溯法中，并不是先构造出整棵状态空间树，再进行搜索，而是在搜索过程，逐步构造出状态空间树，即边搜索，边构造。

最优子结构

某个问题的最优解包含其子问题的最优解

最优化原理

不管前面决策时怎样的，以后的决策只与之前决策所决定的当前状态有关

贪心选择性质

问题的整体解，可以通过一系列局部最优的选择，即贪心选择达到

重叠子问题

每次产生子问题不总是新问题，有些子问题被反复计算多次

分治法：

将规模为n的问题分解为规模为k的相对规模更小的子问题，这些子问题相互独立且与原问题相同，求解这k个子问题的解，当仍然不易求出时，可再划分k个子问题，递归直到子问题很容易求出时，将子问题的解合并为规模更大的问题的解，自底向上逐步合并得到原问题的解

贪心策略：

从某一初值出发，以尽可能快的求解更好的解，当达到某一条件时，算法停止

动态规划：  
对与每一步决策，列出所有可能的局部解，在依据相应的判断条件舍弃那些肯定不能得到最优解的局部解，每一步都经过筛选，以

每一步都是最优解来保证全局最优解

回溯法：

在一棵含有问题全部解的状态空间树上进行深度优先搜索，解为叶结点。每到一个结点，判断以该结点为根的子数中是否含有问题的解，若没有则退回父节点，继续下一深度优先搜索。搜索过程是边搜索边构造的