Date: / /

一、0-1背包问题

队列式 vs 优先队列式

1. 限界函数(是否取=)

$up >= bestp$ 可以取等号，因为右子树可能包含最优解。两者相同。

2. 约束函数

判断能否满足背包容量约束：$cw+w_i \le c$

两者相同

3. 结束条件

队列式：队列为空

优先队列：一直持续到找到最优解或者子节点表为空为止

4. 运行次数

优先队列每次都选择最大优先级的节点作为扩展结点，而且当扩展的结点为叶节点时就结束，所以优先队列式运行次数要少一些。

5. 最优解的更新

进入左子树时，若 $cp+p[i] > bestp$，更新 bestp 为 $cp+p[i]$。

6. 最优解的个数

队列式：可能存在多个最后相同的最优解，因为能够到达的叶节点都会被对比更新

优先队列式：只会有一个最优解，由于最终只会到达一个叶节点，到达叶节点即返回 best 值。

二、装载问题

回溯法 VS 分支限界法

	回溯法	分支限界法
限界函数：	$cw+w[i] <= cap$	$cw+w[i] <= cap$
约束函数：	$cw+rw > bestw$	$cw+rw > bestw$
结束条件：	整个子集树遍历完	队列：队列清空 优先队列：遍历到叶子结点
运动次数：	接近 2^n	队列：接近 2^n　优先队列 $< 2^n$

最优解的更新:	每次遍历到叶子结点时进行更新.	每处理一个结点更新一次
最优解的个数:	若限界函数取 $cw+rw>bestw$, 则有多个最优解; 若限界函数取 $cw+rw\geq bestw$, 则有一个最优解	与回溯法相同
其他:	回溯法思想: DFS+剪枝	BFS+剪枝

队列式 VS 优先队列式

	队列式	优先队列式
限界函数:	$cw+rw<bestw$	$cw+rw<bestw$
约束函数:	$cw+w[i]<c1$	$cw+w[i]<c1$
结束条件:	队列为空	扩展结点为叶结点或队列为空
运行次数:	接近2^n	≤队列式的运行次数
最优解的更新:	每处理一个节点就会更新一次	与队列式相同
最优解的个数:	若限界函数取 $cw+rw<bestw$, 则有多个最优解; 若限界函数取 $cw+rw\leq bestw$, 则有一个最优解	与队列式相同
其他:	按照入队顺序进行出队操作	按照优先级进行出队操作

三、单源最短路径问题

回溯法 VS 分支限界法

约束函数:	能与上一个路径的末端节点相连	同
结束条件:	节点全部被标记	同
运行次数:	回溯法大于优先队列的分支限界	
最优解的更新:	每到达一个节点就会执行更新	
最优解的个数:	回溯大于等于优先队列的分支限界	

Date: / /

其他：复杂度：回溯法与队列式相近，通常大于优先队列式

队列式 VS 优先队列式

	队列式	优先队列式
结束条件：	节点的队列为空	节点全部被标记
运行次数：	队列式大于等于优先队列	
最优解的更新：	每到达一个节点就会执行更新	同
最优解的个数：	队列式大于等于优先队列式	
其他：复杂度：	优先队列式通常优于队列式	

四、TSP问题

回溯法VS分支限界法

	回溯法	分支限界法
限界函数：	当前费用小于当前最优费用	同
约束函数：	两个点之间是否有边相连	同
结束条件：	走完所有可行解后	当叶节点成为扩展节点时
运行次数：	n!	n
最优解的更新：	若当前可行解费用小于当前最优费用则更新	同
最优解的个数：	可能有多个	一个

队列~~式~~式 VS 优先队列式

	队列式	优先队列式
限界函数：	当前费用小于当前最优费用	同
约束函数：	两个点之间是否有边相连	同
结束条件：	队列为空	当叶节点成为扩展节点或队列为空
运行次数：	n!	n
最优解的更新：	若当前可行解费用小于当前最优费用则更新	同
最优解的个数：	可能有多个	一个

Date: / /

总结：回溯法和分支限界法都是在问题的解空间树上搜索问题解的算法。回溯法主要利用深度优先搜索策略，通常目标是找到问题的所有可行解；分支限界法主要利用广度优先搜索策略，通常目标是尽快找到一个满足问题约束条件的解。