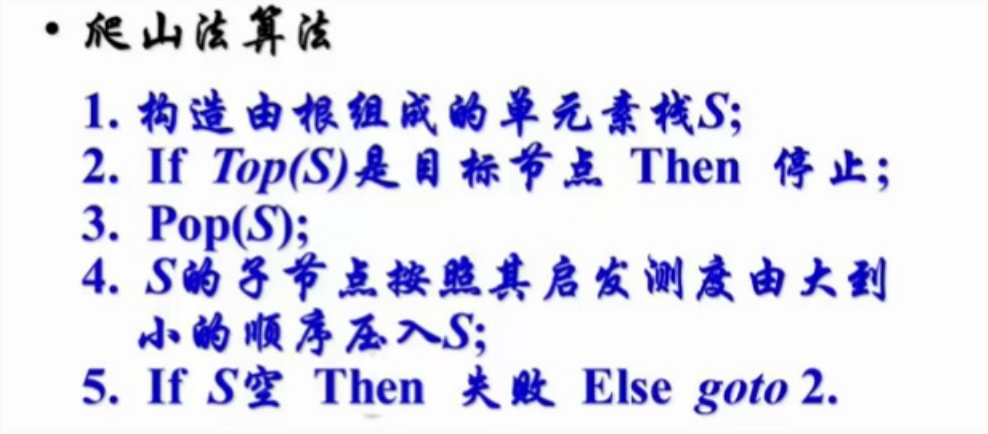
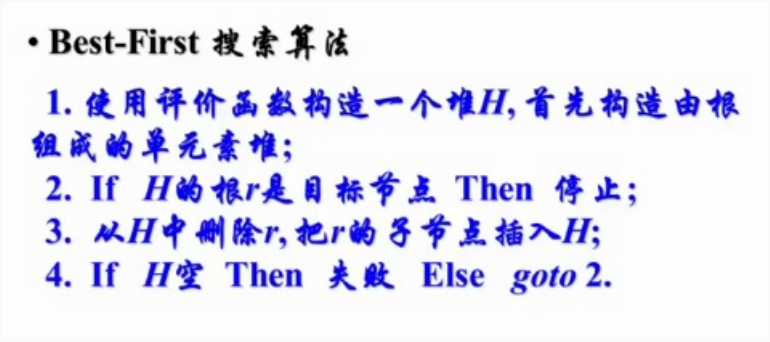
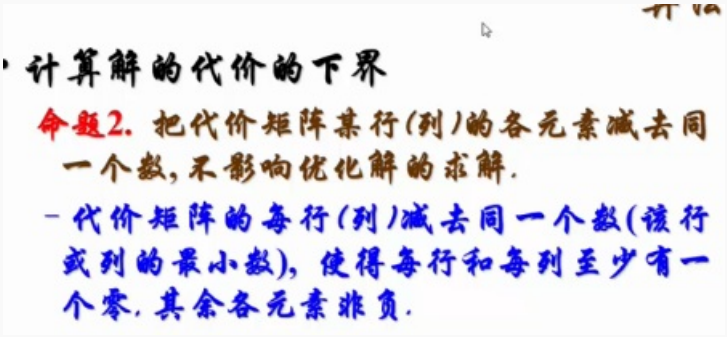
爬山法（深度优先的扩展）
重点：按启发测度从大到小的顺序压入
缺点：有时会陷入局部极值



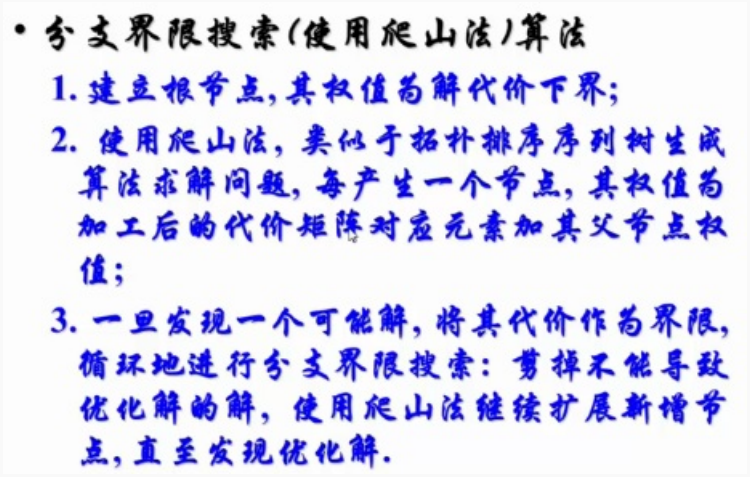
Best-First（广度优先的扩展）
重点：评价函数，使用堆(其实使用最小优先队列)



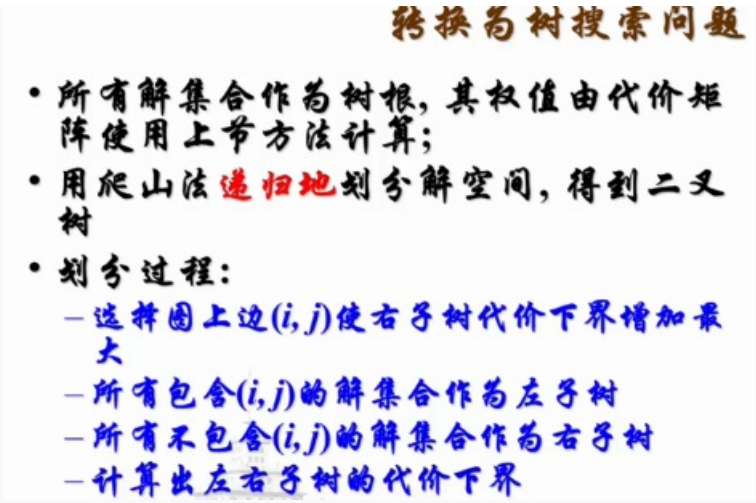
分支界限
重点：利用一定条件进行剪枝，减少搜索空间



解空间的下界就是减去的公共部分的总和

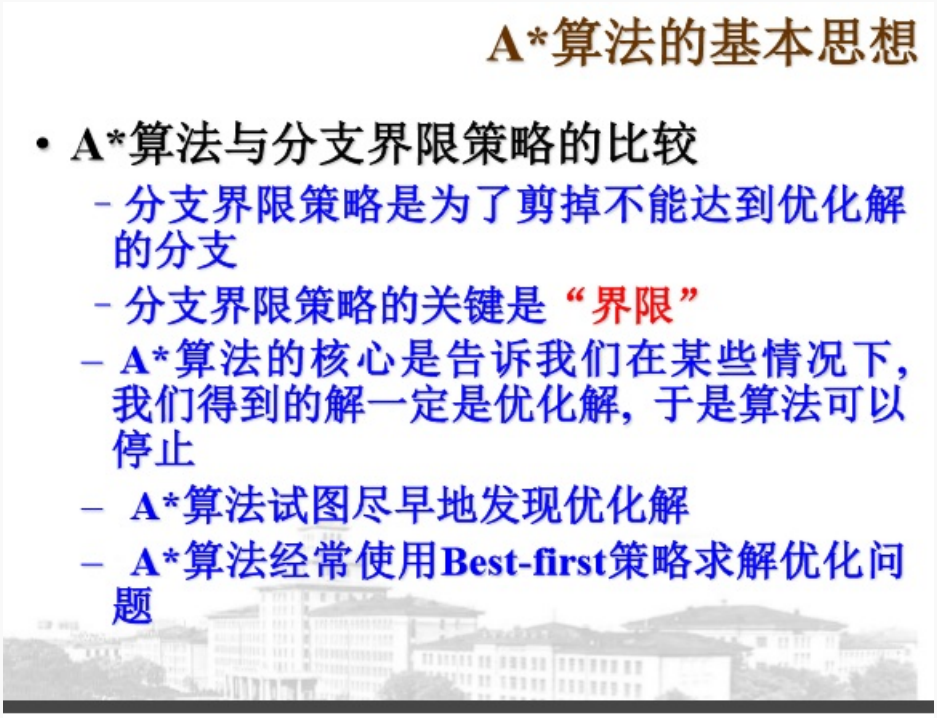


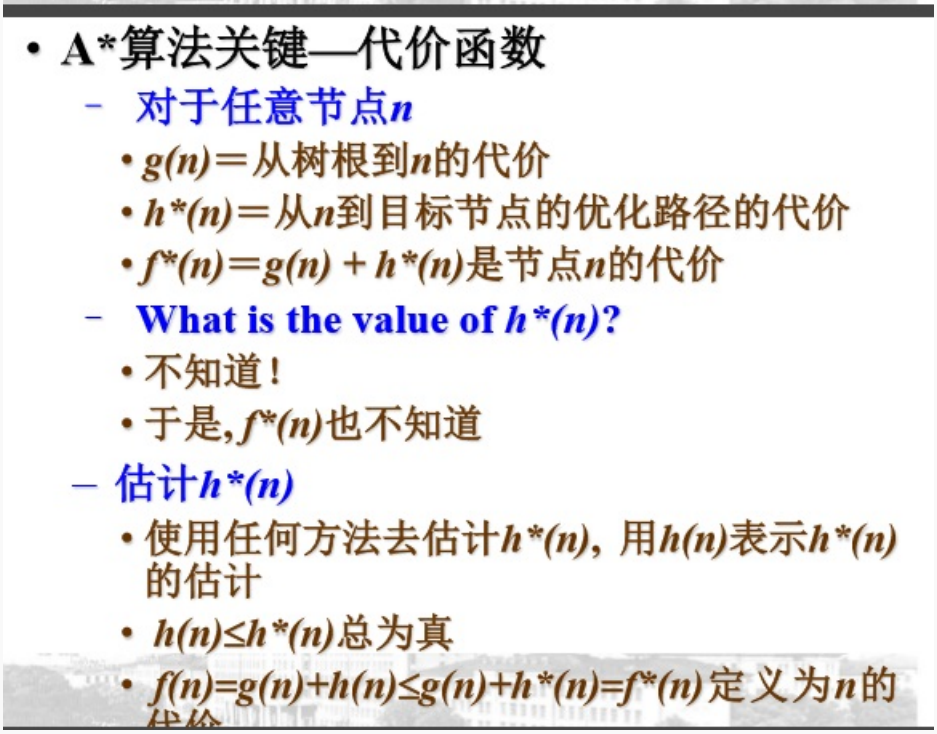
TSP问题



A*

A*算法计算过程中无需剪枝？





- A*算法本质—已经发现的解是优化解

定理1. 使用Best-first策略搜索树，如果A*选择的节点是标节点，则该节点表示的解是优化解.

证明. 令n是任意扩展到的节点，t是选中目标节点.

往证$f(t)=g(t)$是优化解代价.

(1). A*算法使用Best-first策略，$f(t)\leq f(n)$.

(2). A*算法使用$h(n)\leq h^*(n)$估计规则，$f(t)\leq f(n)\leq f^*(n)$.

(3). $\{f^*(n)\}$中必有一个为优化解的代价，令其为$f^*(s)$. 我们有$f(t)\leq f^*(s)$.

(4). t是目标节点$h(t)=0$，所以$f(t)=g(t)+h(t)=g(t)\leq f^*(s)$.

(5). $f(t)=g(t)$是一个可能解，$g(t)\geq f^*(s)$，$f(t)=g(t)=f^*(s)$.

A*算法的规则

(1). 使用Best-first策略搜索树;

(2). 节点n的代价函数为$f(n)=g(n)+h(n)$，$g(n)$是从根到n的路径代价，$h(n)$是从n到某个目标节点的优化路径代价;

(3). 对于所有n，$h(n)\leq h^*(n)$;

(4). 当选择到的节点是目标节点时，算法停止，返回一个优化解.