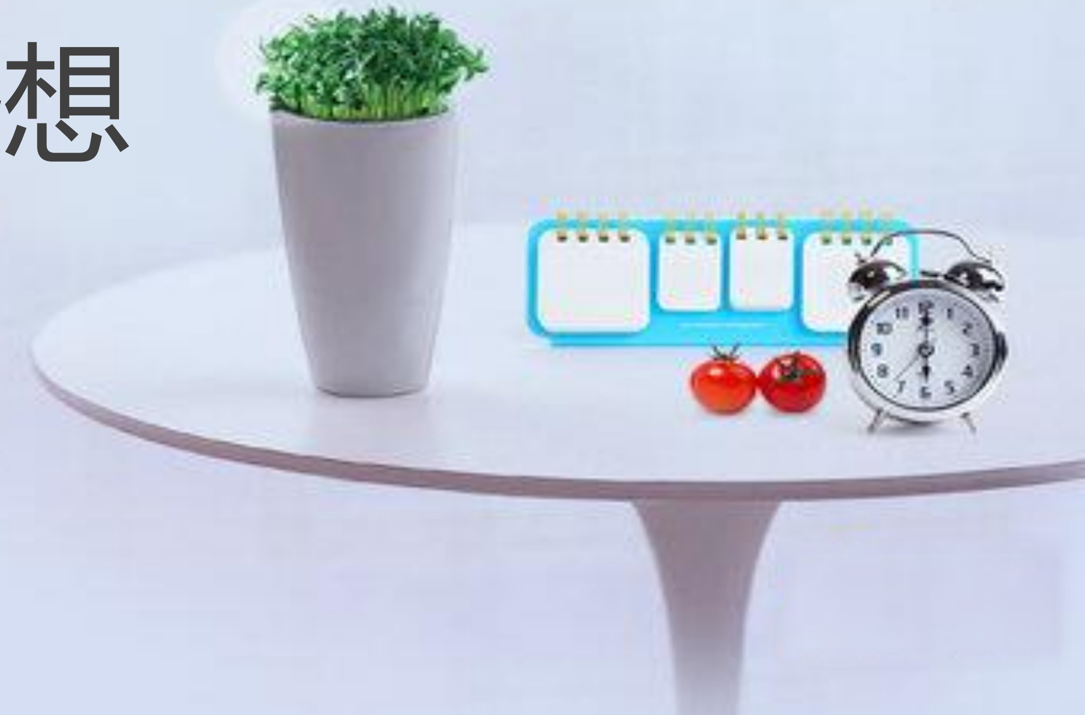
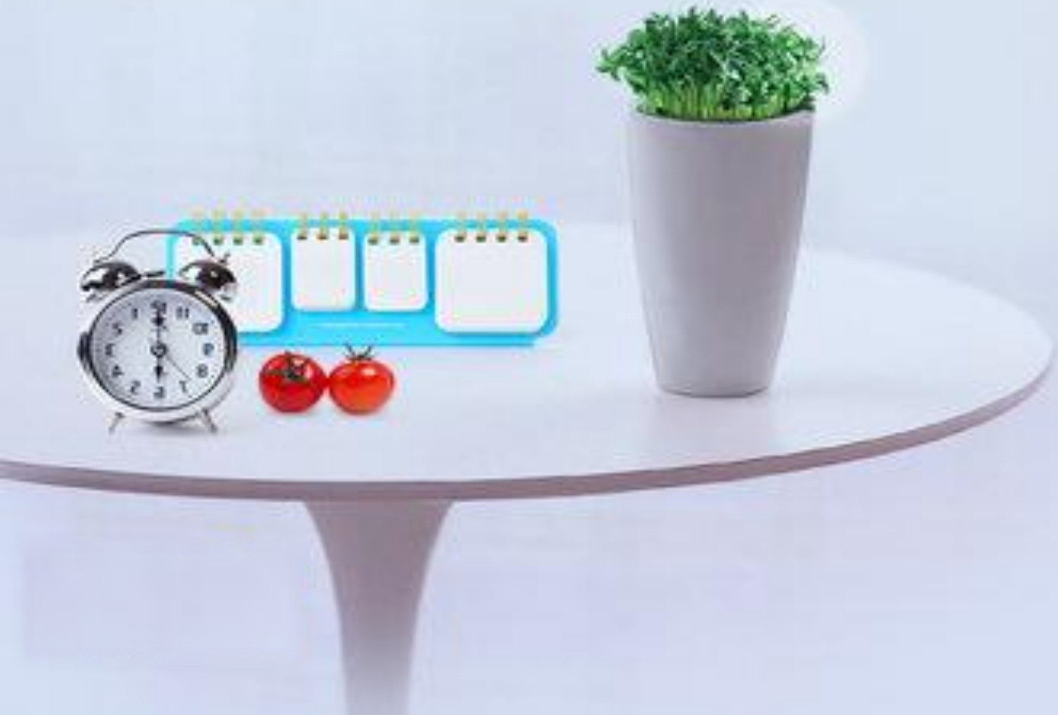
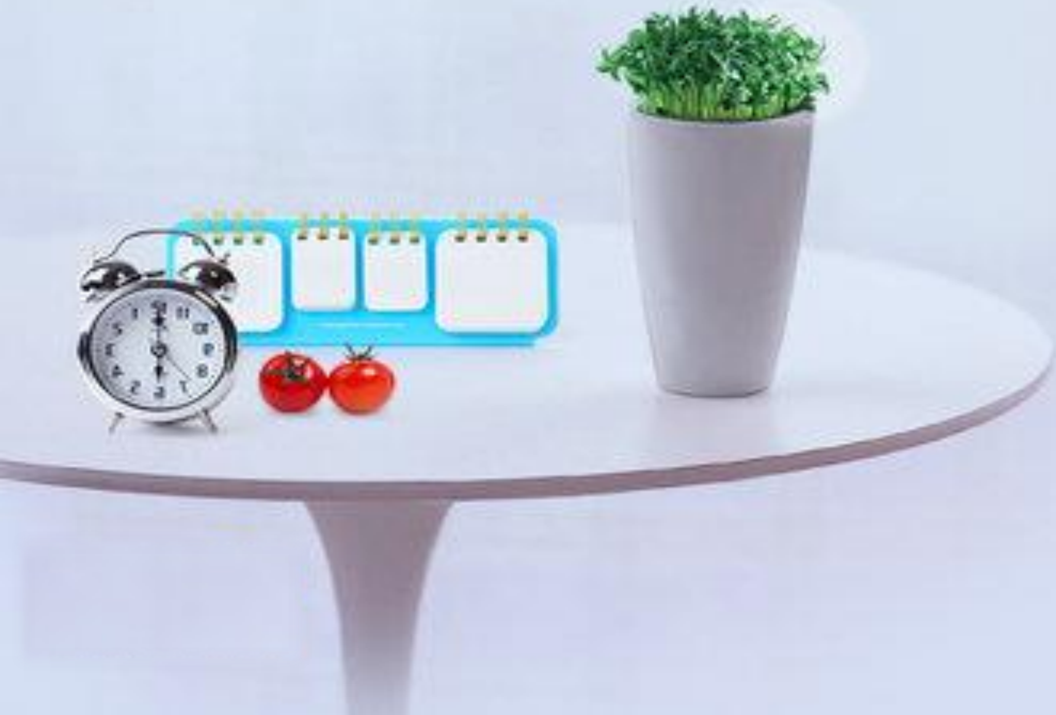
搜索算法综述与论文感想

学号：1951976　　　　姓名：李林飞





目录



01

搜索算法综述

01 盲目搜索-BFS

Breadth-first search

完备性

完备

最优性

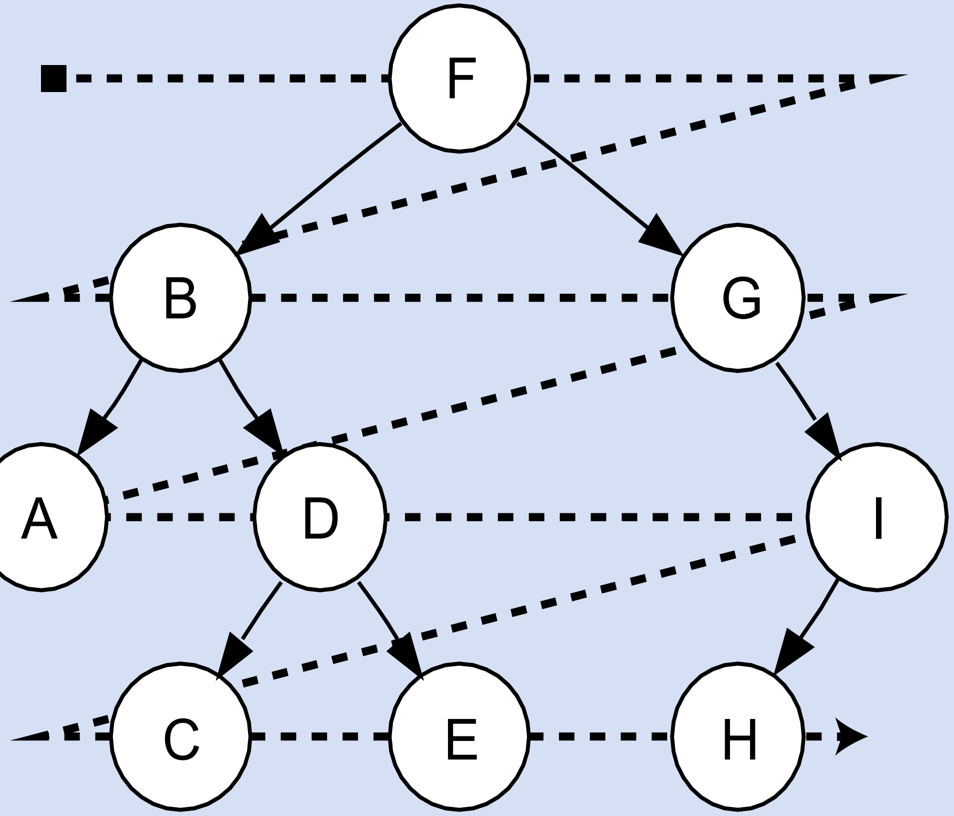
只有满足路径代价是基于结点深度的非递减函数时，BFS 算法才满足最优性。

时间复杂度

$O(b^d)$

空间复杂度

$O(b^d)$



01

盲目搜索-DFS

Depth First Search

完备性

有限图搜索：完备
树搜索：不完备

最优性

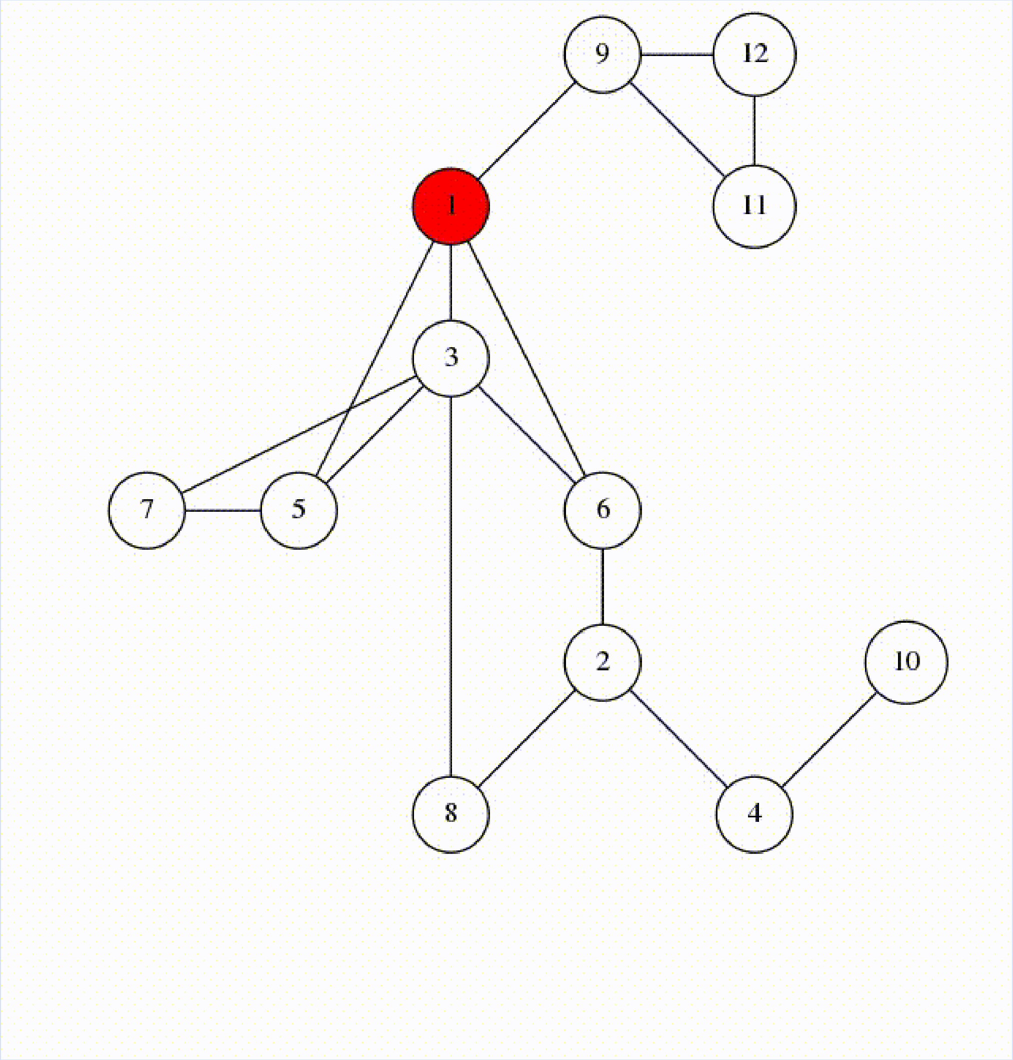
无论是图搜索还是树搜索，都不是最优的

时间复杂度

$$O(b^m)$$

空间复杂度

$$O(bm)$$



01 盲目搜索-IDDFS

Iterative deepening depth-first search

完备性

有深度限制，是完备的

最优性

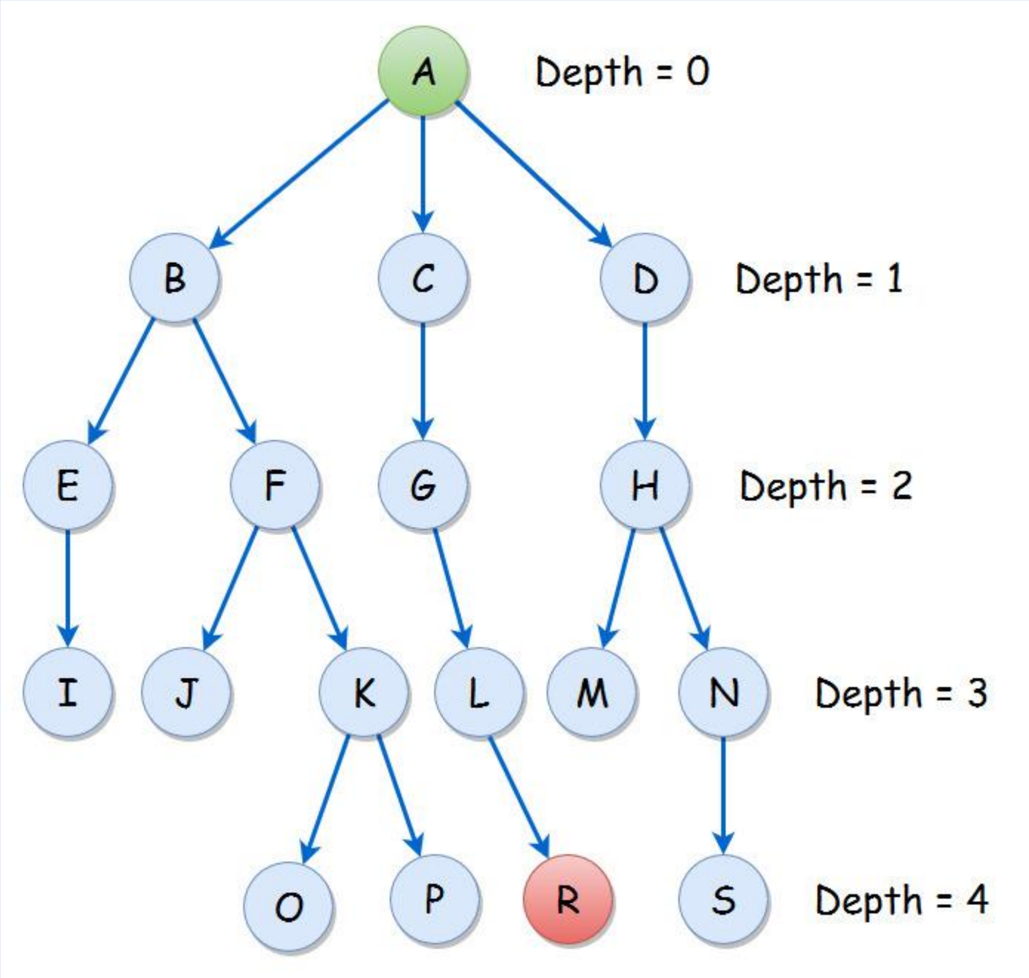
只有在路径代价是结点深度的非递减函数时该算法才是最优的。

时间复杂度

$$O(b^d)$$

空间复杂度

$$O(bd)$$



02 启发式搜索-贪婪最佳优先搜索

Greedy Best-First Search

完备性

在有限空间上，图搜索是完备的；树搜索在有限状态下也是不完备的

最优性

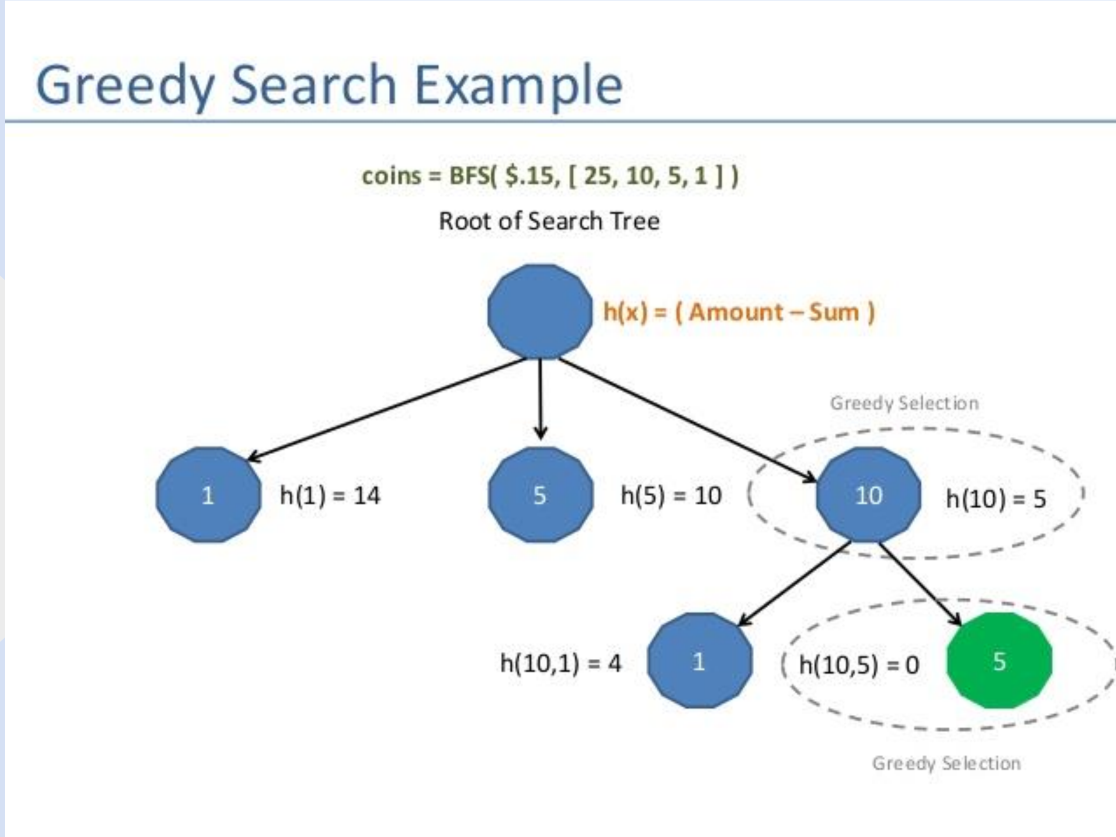
不是最优的

时间复杂度

$$O(b^m)$$

空间复杂度

$$O(b^m)$$



02 启发式搜索-A*

A Star

完备性

如果在搜索的过程中每步代价都超过从根结点到达目标结点的估计代价与实际代 价的相对误差并且 b（任意结点的最多后继数）是有穷的，则 A* 算法是完备的。

最优性

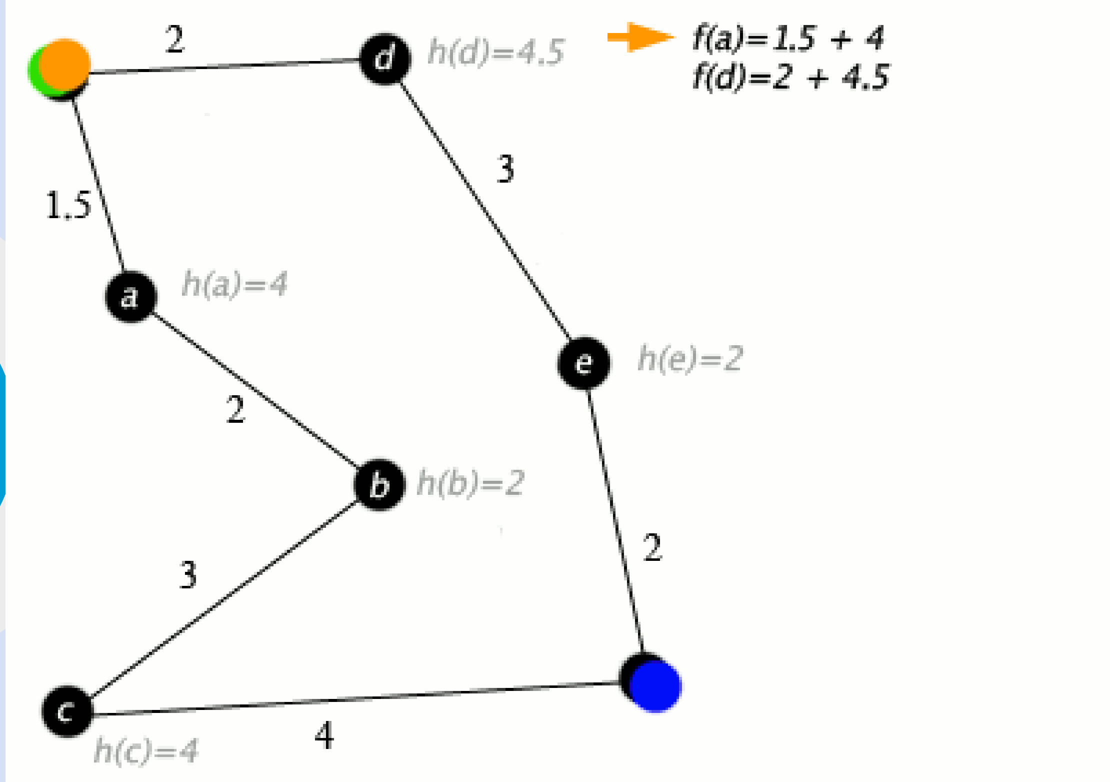
如果 h(n) 是可采纳的，那么 A* 的树搜索版本是最优的；如果 h(n) 是一致的，那 么图搜索的 A* 算法是最优的。

时间复杂度

严重依赖于对状态空间的所做的假设

空间复杂度

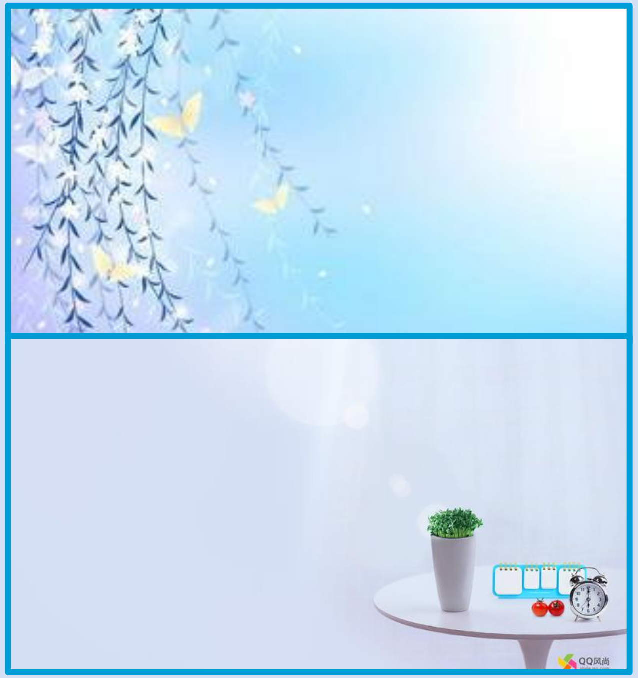
严重依赖于对状态空间的所做的假设

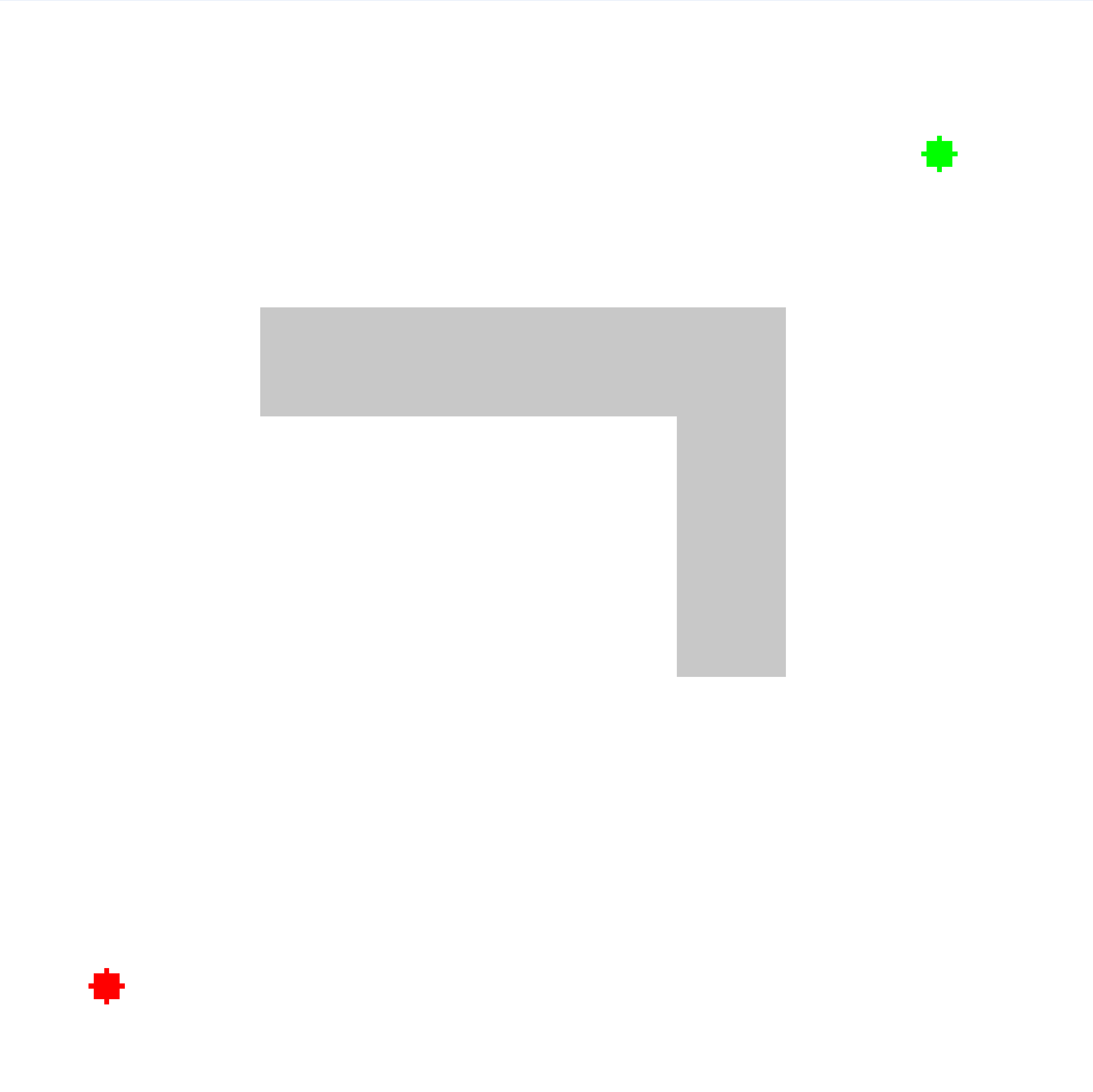


02

启发式搜索—A*

A Star





03 局部搜索-模拟退火

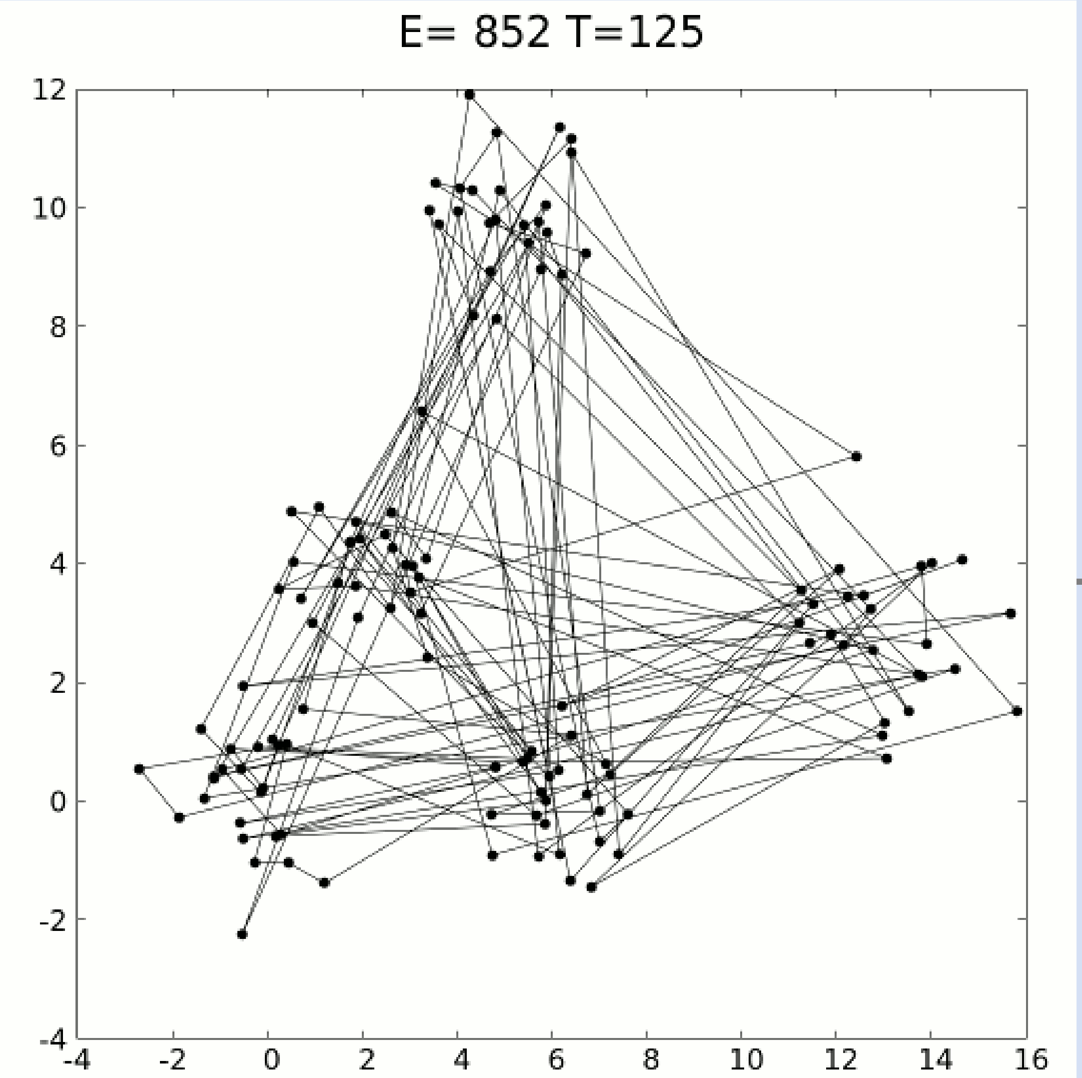
Simulated Annealing

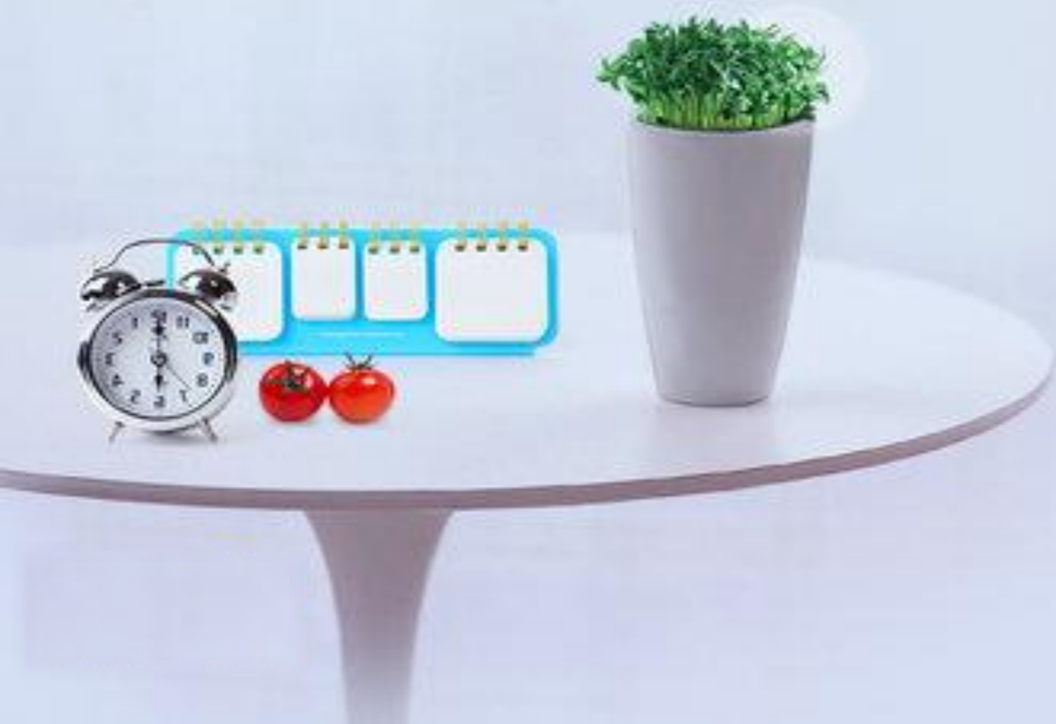
在爬山法中加入随机因素的 SA 算法是完备的，因为从后继集合中完全等概率的随 机选取后继是可以找到解的，但是效率会很低。

如果调度随机移动的距离，使其下降的足够慢，则算法找到全局最 优解的概率就会逼近 1。

完备性

最优性





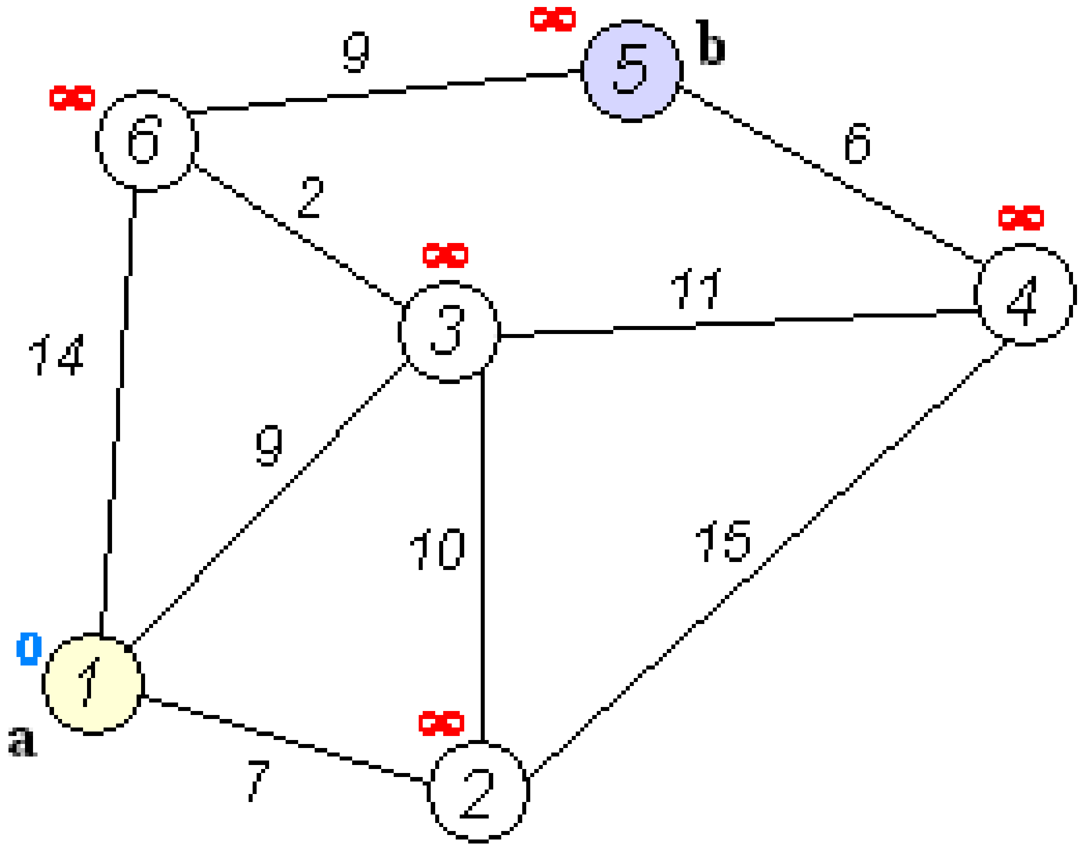
02

论文阅读感想

01 Dijkstra算法

Dijkstra Algorithm

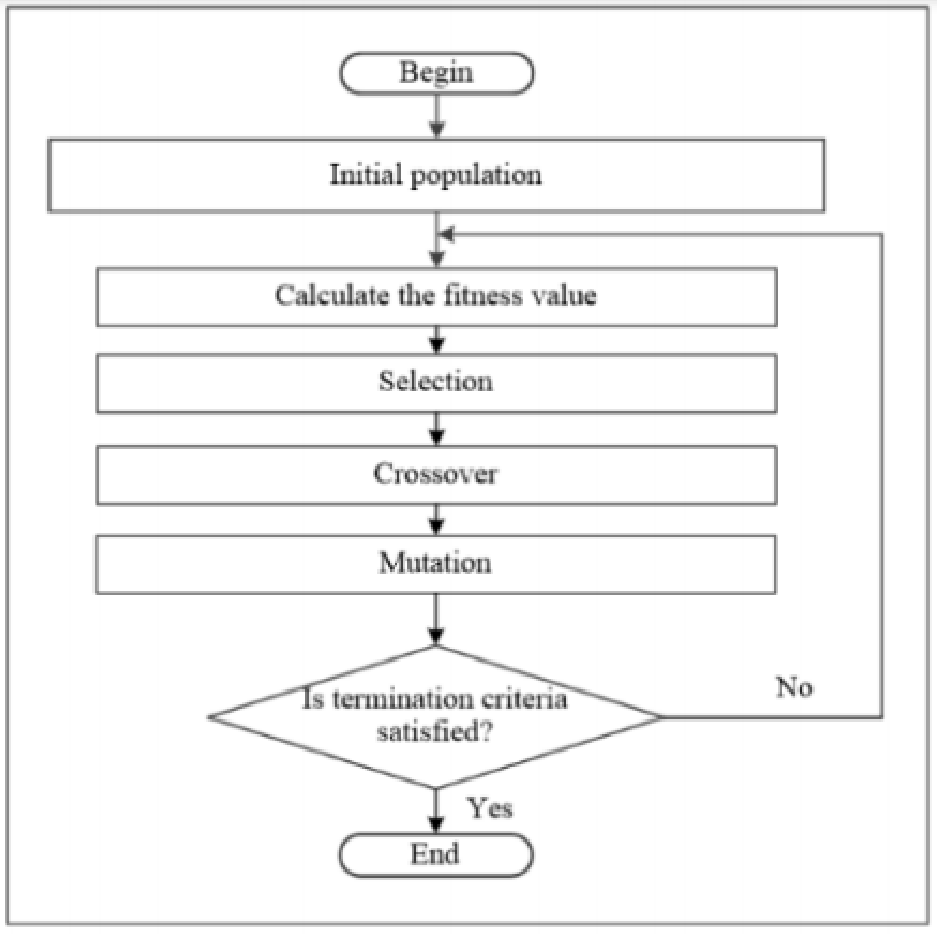
```
function Dijkstra(Graph, source):

    create vertex set Q

    for each vertex v in Graph:
        dist[v] ← INFINITY
        prev[v] ← UNDEFINED
        add v to Q
    dist[source] ← 0

    while Q is not empty:
        u ← vertex in Q with min dist[u]

        remove u from Q

        for each neighbor v of u:           // only v that are still in Q
            alt ← dist[u] + length(u, v)
            if alt < dist[v]:
                dist[v] ← alt
                prev[v] ← u

    return dist[], prev[]
```



02

遗传算法

Genetic Algorithm



02 TSP模型化

Travel Salesman Problem

设 $G=(V,E)$ 为赋权图，$V=1,2,...,n$ 为顶点集，E 为边集，各顶点间距离为 C_{ij}，已知 $(C_{ij}>0,i,j\in V)$, 并设

$$x_{ij}=\begin{cases}1 & \text{在最优路径上}\\ 0 & \text{others}\end{cases}$$

则 TSP 问题可用如下数学模型 (1)-(5) 式联立表示：

$$minZ=\sum_{i\neq j}c_{ij}x_{ij} \tag{1}$$

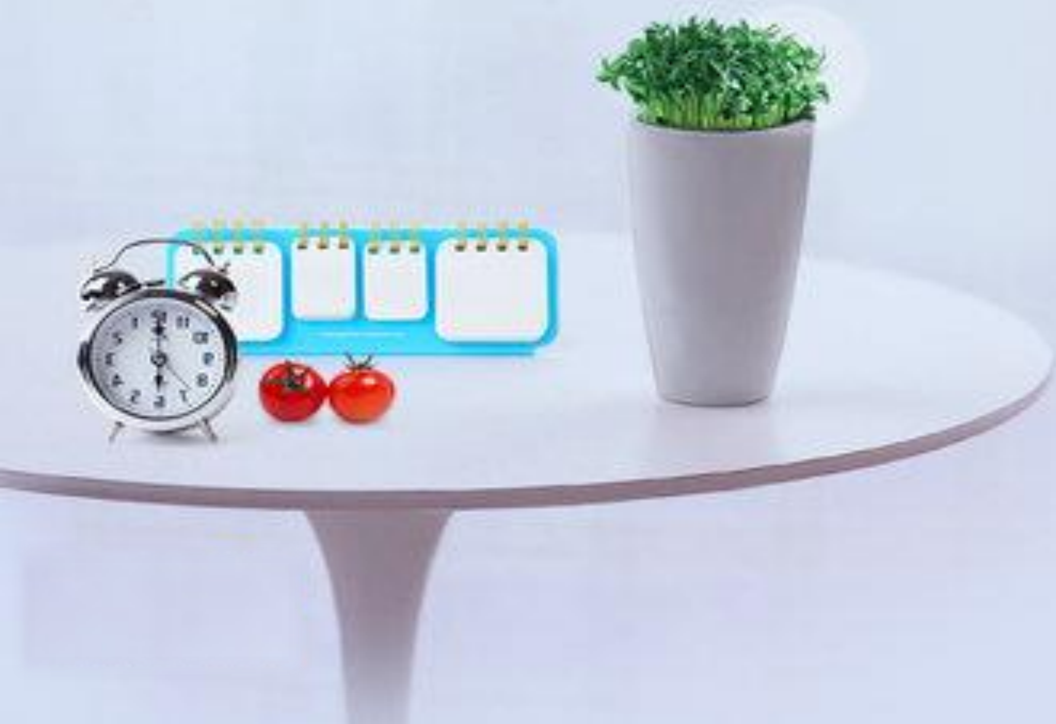
$$\sum_{i\neq j}x_{ij}=1 \quad j\in v \tag{2}$$

$$\sum_{j\neq i}x_{ij}=1 \quad j\in v \tag{3}$$

$$\sum\nolimits_{i,j\in s}x_{ij}\leq|K|-1 \quad k\in v \tag{4}$$

$$x_{ij}\in 0,1 \quad i,j\in v \tag{5}$$

其中，K 是 V 的全部非空子集，$|K|$ 是集合 K 中包含图 G 的全部顶点的个数。



03

启发与思考

Ideas And Methods

过程	二叉堆（最坏情况）	二项堆（最坏情况）	斐波那契堆（平摊）
MAKE-HEAP	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
INSERT	$\Theta(\lg n)$	$\Omega(\lg n)$	$\Theta(1)$
MINIMUM	$\Theta(1)$	$\Omega(\lg n)$	$\Theta(1)$
EXTRACT-MIN	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$
UNION	$\Theta(n)$	$\Omega(\lg n)$	$\Theta(1)$
DECREASE-KEY	$\Theta(\lg n)$	$\Theta(\lg n)$	$\Theta(1)$
DELETE	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$

01 理性认识-分析能力

对于Dijkstra算法，给出时间复杂度的计算公式

$$(n-1)*(T_{EXTRACT-MIN}+T_{DELETE}+T_{DECREASE-KEY}*k)$$

顺序遍历集合

$$\begin{aligned}
&=(n-1)*(T_{EXTRACT-MIN}+T_{DELETE}+T_{DECREASE-KEY}*k)\\
&=(n-1)*(n+1+k)\\
&=n*(n+k)\\
&=n^2+m
\end{aligned}$$

二叉堆作为优先队列

$$\begin{aligned}
&=(n-1)*(T_{EXTRACT-MIN}+T_{DELETE}+T_{DECREASE-KEY}*k)\\
&=(n-1)*(1+\log n+k*\log(n))\\
&=n*(1+k)\log n\\
&=(n+m)\log n
\end{aligned}$$

二项堆作为优先队列

$$\begin{aligned}
&=(n-1)*(T_{EXTRACT-MIN}+T_{DELETE}+T_{DECREASE-KEY}*k)\\
&=(n-1)*(\log n+\log n+k*1)\\
&=n*(2\log n+k)\\
&=2n\log n+m\\
&=n\log n+m
\end{aligned}$$

斐波那契堆作为优先队列

$$\begin{aligned}
&=(n-1)*(T_{EXTRACT-MIN}+T_{DELETE}+T_{DECREASE-KEY}*k)\\
&=(n-1)*(\log n+\log n+k*\log(n))\\
&=n*(2+k)\log n\\
&=(2n+m)\log n\\
&=(n+m)\log n
\end{aligned}$$

02 感性认识

Nature is the best teacher!

THANK YOU

学号：1951976 姓名：李林飞