

算法	最大时间	平均时间	最小时间	辅助空间 代价	稳定性
冒泡排序	$O(n^2)$ 换相邻两项	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
改进冒泡排序	$O(n^2)$	$O(n^2)$	$O(1)$	$O(1)$	稳定
快速排序	$O(n^2)$ 将 $a_i$ 排对位置	$O(n\log n)$	$O(n\log n)$	$O(\log n)$	不稳定
基数排序	$O(d*(n+r))$	$O(d*(n+r))$	$O(d*(n+r))$	$O((n+r))$	稳定
归并排序	$O(n\log n)$ == 分治	$O(n\log n)$	$O(n\log n)$	$O(n)$	稳定

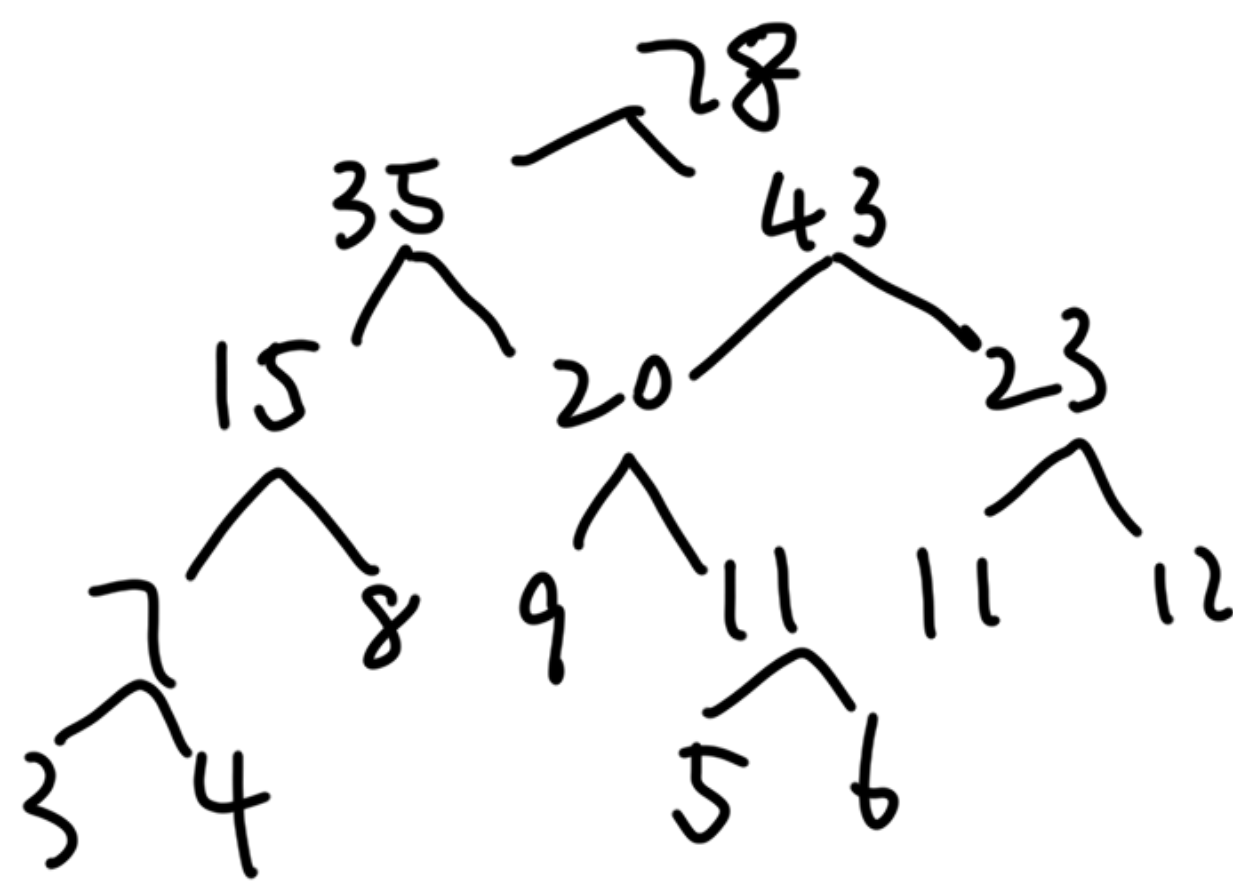
算法	最大时间	平均时间	最小时间	辅助空间代 价	稳定性
直接插入排序	$O(n^2)$ 前 $n$ 轮排出前 $n$ 个数	$O(n^2)$	$O(n)$	$O(1)$	稳定
二分插入排序	$O(n^2)$	$O(n^2)$	$O(n\log n)$	$O(1)$	稳定
Shell排序	$O(n^{1.3})$ 按 $2^t$ 分组	$O(n^{1.3})$	$O(n^{1.3})$	$O(1)$	不稳定
直接选择排序	$O(n^2)$ 换 $a_i$ 与 $a_{min}$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
堆排序	$O(n\log n)$ 建堆 $O(n)$	$O(n\log n)$	$O(n\log n)$	$O(1)$	不稳定

前序：根左右

中序：左根右

后序：左右根

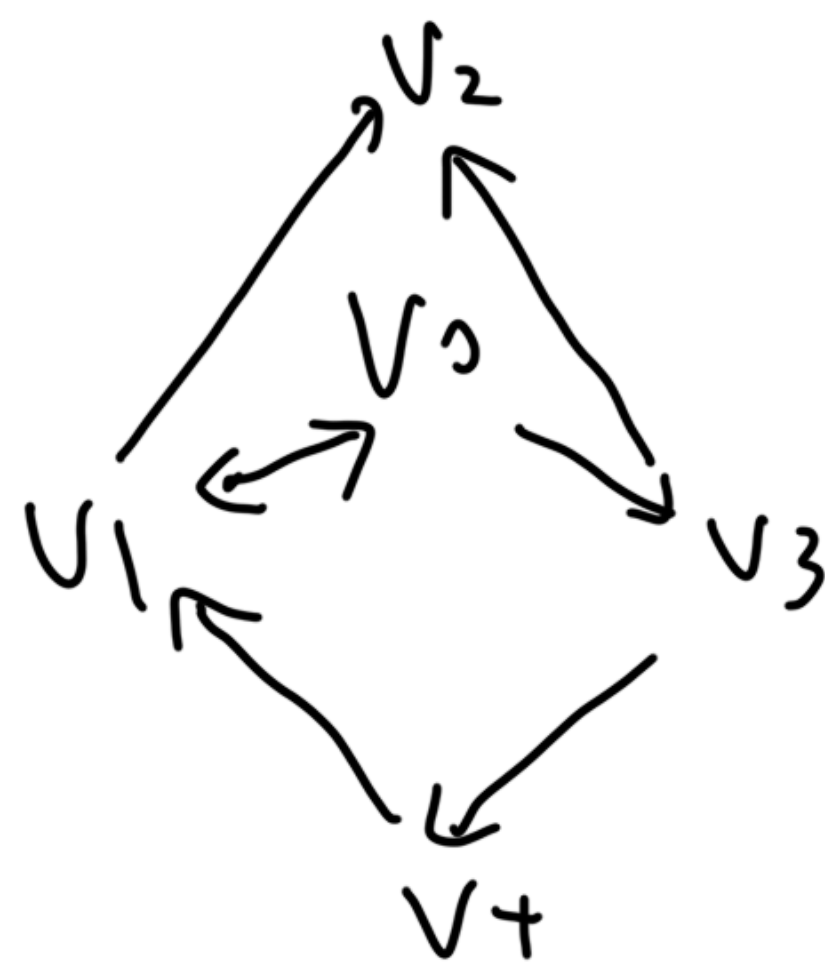
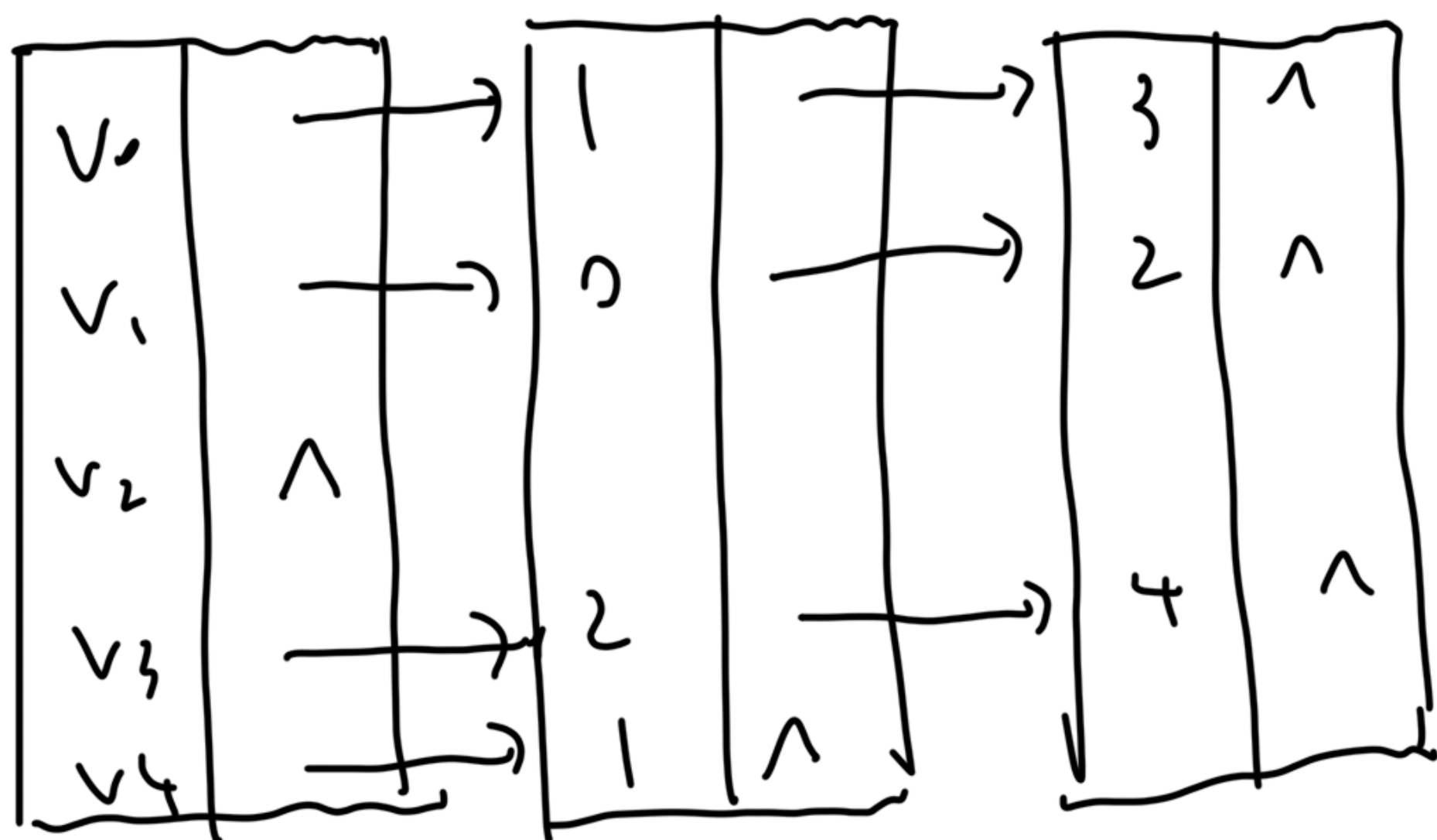
Huffman 算法：{3, 4, 5, 6, 8, 9, 11, 12} = 5



再按左小右大

最优二叉编码树，权外部路径长

为  $\sum_i i \cdot w_i$



拓扑序列：(AOV网)：

按顺序压入无入度点  
出栈 A，删 A 所有边

交替

二叉搜索树

左节点 < 根 < 右节点

无重复值



(二叉排序树)

平衡二叉树 (AVL树)

左右树深度差  $\leq 1$

满二叉树 :

节点为叶节点 / 叶子节点

完全二叉树

(堆)

AVL树

最小生成树

① Prim 算法

以A出发, 依次加入与集合

连通的最短边

② Kruskal

加入最短可行边

散列表检索效率: 散列函数, 碰撞处理方法, 装填因子.

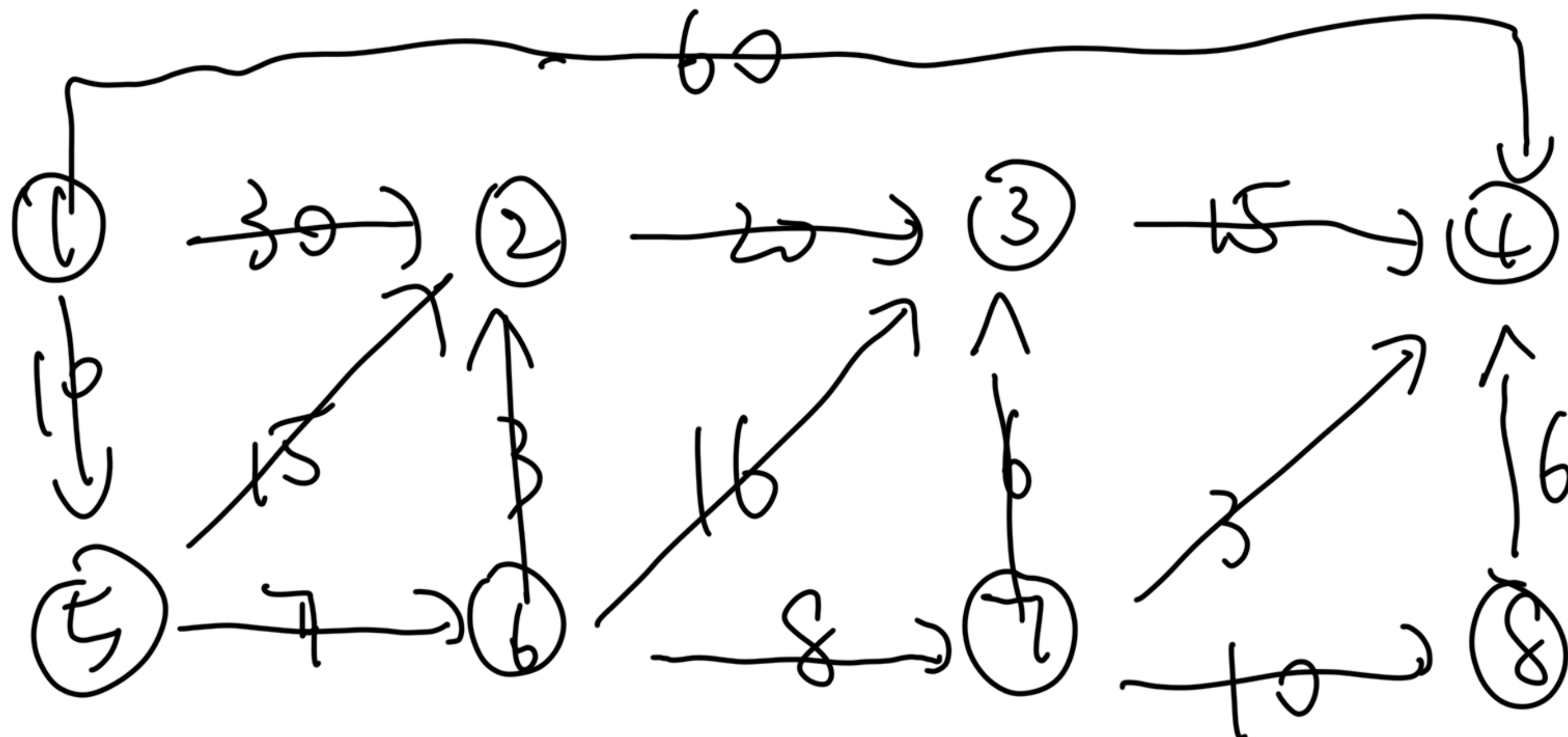
以列A指针  $[0, m]$  F.R , 牺牲一个空间分空溢,  $(R+1) \% (m+1) = F$ .

NEXT<sub>1</sub>:    a b c a b b c  
 $[1 \ 0 \ 0 \ 0 \ 1 \ 2 \ 0]$   
                   ↑  
                   "ab" = "ab"

NEXT<sub>2</sub>:  
 $(\begin{smallmatrix} a & b & c & a & b & b & c \\ 1 & 0 & 0 & 0 & 1 & 2 & 0 \end{smallmatrix}) \rightarrow (\begin{smallmatrix} \underline{a} & \underline{b} & c & a & \underline{b} & b & c \\ 1 & 0 & 0 & 0 & 1 & 1 & 2 & 0 \end{smallmatrix})$   
                   Δ                  Δ                  Δ                  Δ

a b c a b b c →  $(\begin{smallmatrix} a & b & c & a & b & b & c \\ 1 & 0 & 0 & 1 & 0 & 2 & 0 \end{smallmatrix})$

Dijkstra:



1 1 }

30, ∞, 60, 10, ∞, ∞, ∞

1 1.5 }

25, ∞, 60, 10, 17, ∞, ∞

1 1.5.6 }

20, 33, 60, 10, 17, 25, ∞