

# 基于 HMM 的中文分词

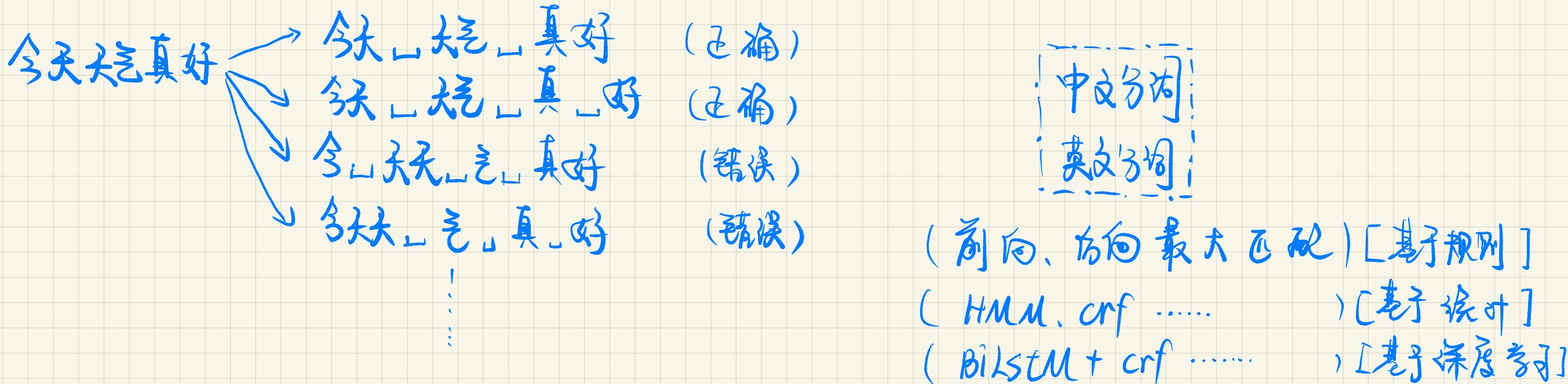
手写 AI



扫一扫上面的二维码图案，加我微信

- 01/ L 中文分词任务 1
- 02/ L 语料库 1
- 03/ L 初始、转移、发射矩阵 1
- 04/ L 维特比算法 1
- 05/ L 代码解读 1

# 中文分词



## 为什么要分词

- ① 更好的理解语义
- ② 为了更重要的任务  
(命名实体识别、情感分析、文本分类、语义识别.....)
- ③ 应用场景需要  
(淘宝、百度....)

并非所有中文任务都需要分词

# 资料库 (名词)

是全国各族人民在中国共产党领导下，在建设有中国特色的社会主义道路上，坚持改革、开放，团结奋斗、胜利前进的一年。城乡经济体制改革向纵深稳步推进，对外开放迈出了新的步伐，工农业生产和其它各项建设事业全面完成了“七五”人民生活继续有所改善。政治上安定团结，端正党风和社会风气的工作取得了新的进展，社会主义民主和法制建设不断加强。在党的十二届六中全会通过的《关于社会主义精神文明建设若干问题的决议》，我国两个文明的建设正在向新的水平迈进。从党的十一届三中全会实现伟大历史转折到现在，我国政治安定团结，经济稳定、持续、协调发展已经八年了，这是建国以来稳步发展持续时间最长的时期。在十年动乱之后，取得这样一个大好局面是不容易的。

- ①每一段是一篇“文章”；
- ②每篇文章用空格分开；
- ③资料库的准确性，严重影响写词结果；
- ④理论上，资料库越大越好

每个字都有一个标记，“隐藏状态”  
可以根据资料库得到所有标记，

B：词语开始

M：词语中间

E：词语结束

S：单独成词

中文多词就是为了得到节奏：

麻辣肥牛真好吃！

B M M E S B E S ↓

根据已知状态进行分词。  
即在“E”和“S”后面划  
出空格即可：

BMME <sub>4</sub> S <sub>2</sub> BE <sub>2</sub> S

麻辣肥牛 真 好吃！

已分好的词 ⇒ 每个字的状态

是全国各族人民在中国共产党领导下，在建设有中国特色的社会主义道路上，坚持改革、开放，团结奋斗、胜利前进的一年。城乡经济体制改革向纵深稳步推进，对外开放迈出了新的步伐，工农业生产和其它各项建设事业全面完成了“七五”人民生活继续有所改善。政治上安定团结，端正党风和社会风气的工作取得了新的进展，社会主义民主和法制建设不断加强。在党的十二届六中全会通过的《关于社会主义精神文明建设两个文明的建设正在向新的水平迈进。从党的十一届三中全会实现伟大历史转折到现在，我国政治安定团结，经济稳定、持续、协调发展已经八年了，这是建国以来稳步发展持续时间最长的时期。在十年动乱之后，取得这样一个大好局面是不容易的。

S B E B E B E S S B E B M E B E S S  
S B E S B E B E S S B M M E B E S S  
B E B E S B E S  
B E B E S B E B E S S S S



# HMM 与词训练与预测

今天天气真不错。  
麻辣肥牛 好吃！  
我喜欢 吃 好吃的！



B E B E S B E S  
B M M E B E S  
S B E S B E S S



{ 初始概率矩阵  
转移概率矩阵  
发射矩阵

训练

今天的天气不错

{ 初始概率矩阵  
转移概率矩阵  
发射矩阵

计算所有可能性的概率

B	M	E	B	E	S	S
S	S	B	E	B	E	S
S	B	E	B	M	M	E
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

→ B E S B E B E

维特比算法

预测

初始矩阵：

统计通篇文章的第一个字是什么状态

(一开始统计的数值都是频次)

B	M	S	E
2	0	1	0



B	M	S	E
0.667	0	0.333	0

转移矩阵：

当前状态到下一状态的概率

	B	M	S	E
B	0	1	0	6
M	0	1	0	1
S	3	0	1	0
E	2	0	5	0



	B	M	S	E
B	0	0.142	0	0.857
M	0	0.5	0	0.5
S	0.75	0	0.25	0
E	0.285	0	0.715	0

味道真不错。  
麻辣肥牛好吃！  
我喜欢吃好吃的！

BEBESBES  
BMMEBES  
SBESESS

发射矩阵：统计某种状态下，所有字出现的次数（概率）

B	今:1	天:1	不:1	麻:1	好:2	喜:1
M	辣:1	肥:1				
S	。:1	!:2	我:1	吃:1	风:1	真:1
E	大:1	错:1	牛:1	吃:2	欢:1	气:1



B	今:0.142	天:0.142	不:0.142	麻:0.142	好:0.285	喜:0.142
M	辣:0.5	肥:0.5				
S	。:0.142	!:0.285	我:0.142	吃:0.142	风:0.142	真:0.142
E	大:0.142	错:0.142	牛:0.142	吃:0.285	欢:0.142	气:0.142

## 初始矩阵：

统计每篇篇章的第一个字是什么状态

(一开始统计的数值都是次数)

B	M	S	E
2	0	1	0



B	M	S	E
0.667	0	0.333	0

## 转移矩阵：当前状态到下一状态的概率

	B	M	S	E
B	0	1	0	6
M	0	1	0	1
S	3	0	1	0
E	2	0	5	0



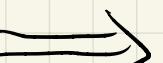
	B	M	S	E
B	0	0.142	0	0.857
M	0	0.5	0	0.5
S	0.75	0	0.25	0
E	0.285	0	0.715	0

钛链真不错。  
麻辣肥牛好吃！  
我喜欢吃好吃的！

B E B E S B E S  
B M M E B E S  
S B E S B E S S

## 发射矩阵：某一个字对应的状态概率

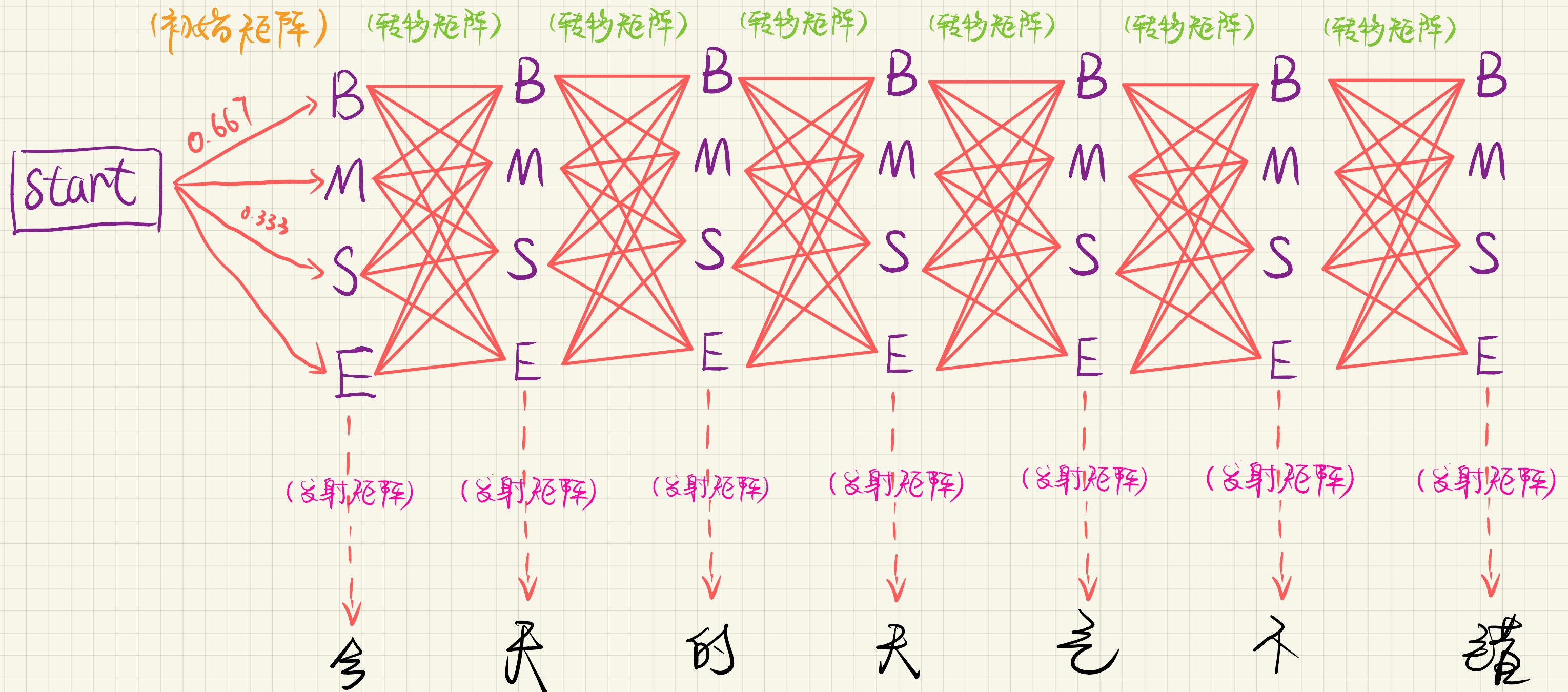
	B	M	S	E
今	1			
天	1			1
气				1
真		1		
不	1			
瑞			1	
:	:	:	:	:



	B	M	S	E
今	1			
天	0.5			0.5
气				1
真		1		
不	1			
瑞			1	
:	:	:	:	:

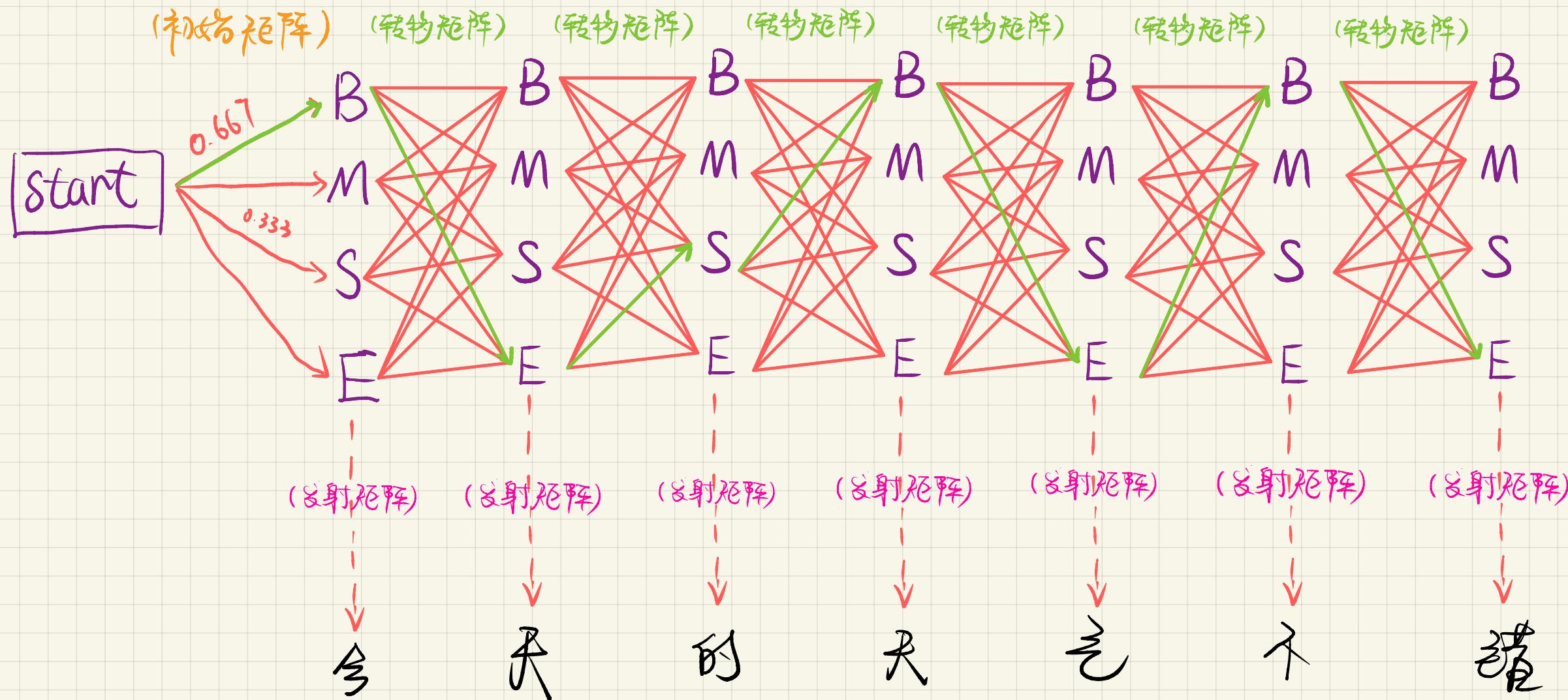
欲速則不

「我的天氣不錯」 $\Rightarrow$  BE S BE BE  $\Rightarrow$  「我的\_天氣\_不錯」



从以上总路程  $4^7$  中选择一条最优路径，即是最终结果。

# 计算



B	M	S	E	
B	0	0.142	0	0.857
M	0	0.5	0	0.5
S	0.75	0	0.25	0
E	0.285	0	0.715	0

初始矩阵

	B	M	S	E
B	0	0.142	0	0.857
M	0	0.5	0	0.5
S	0.75	0	0.25	0
E	0.285	0	0.715	0

转移矩阵

B	今: 0.142	天: 0.142	的: 0.142	是: 0.142	不: 0.142	错: 0.285	真: 0.142
M	辣: 0.5	肥: 0.5					
S	。: 0.142	!: 0.285	升: 0.142	吃: 0.142	网: 0.142	真: 0.142	
E	大: 0.142	错: 0.142	不: 0.142	吃: 0.285	网: 0.142	真: 0.142	

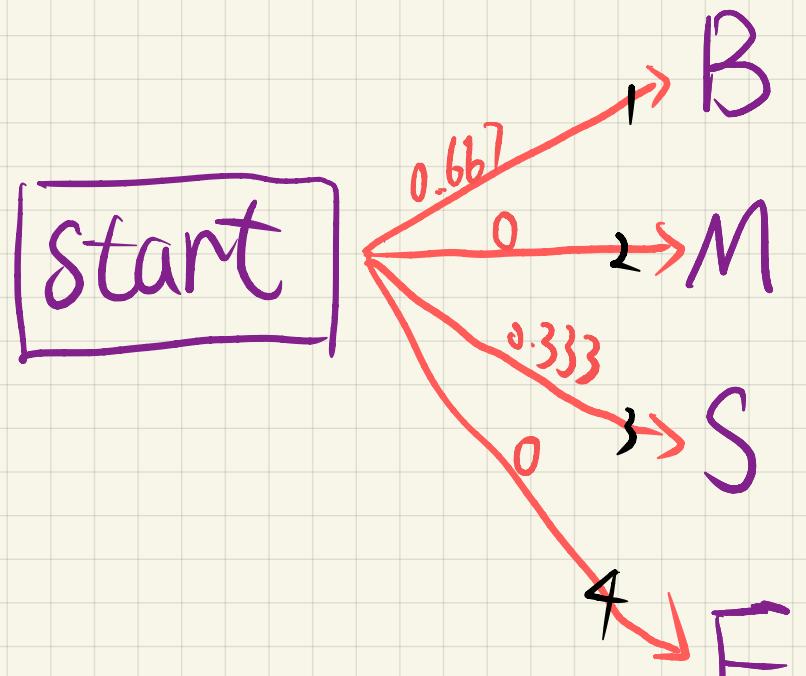
发射矩阵

$$0.667 \times 0.142 \times [0.857 \times 0.142] \times [0.715 \times 0.142] \times [0.75 \times 0.142] \times [0.857 \times 0.142] \times [0.285 \times 0.142] \times [0.857 \times 0.142]$$

今 天 的 是 不 错

# 维特比算法 (从众多路径中, 迅速选择出最优路径)

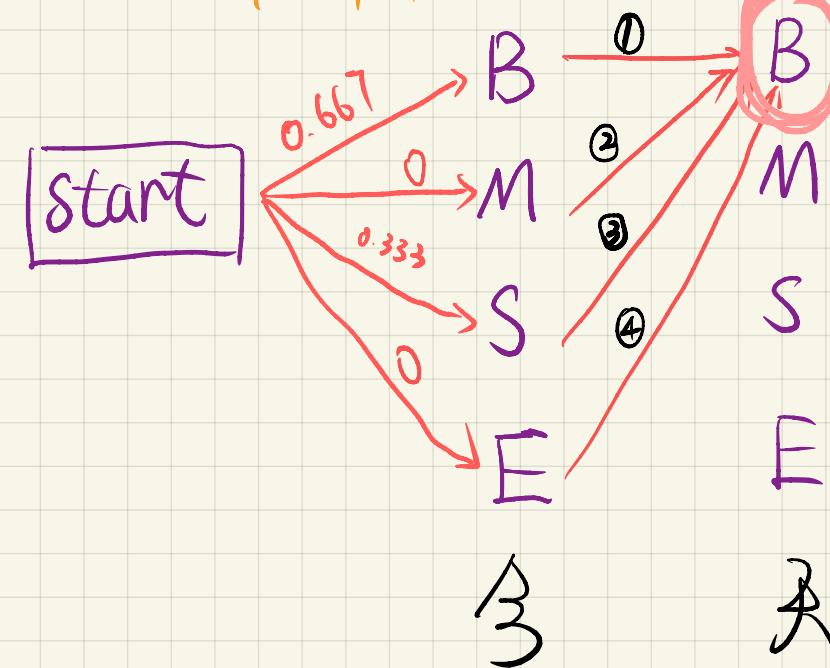
(初始矩阵)



③

(1~4 中的最优但并不一定步全局最优, 都成为候选)

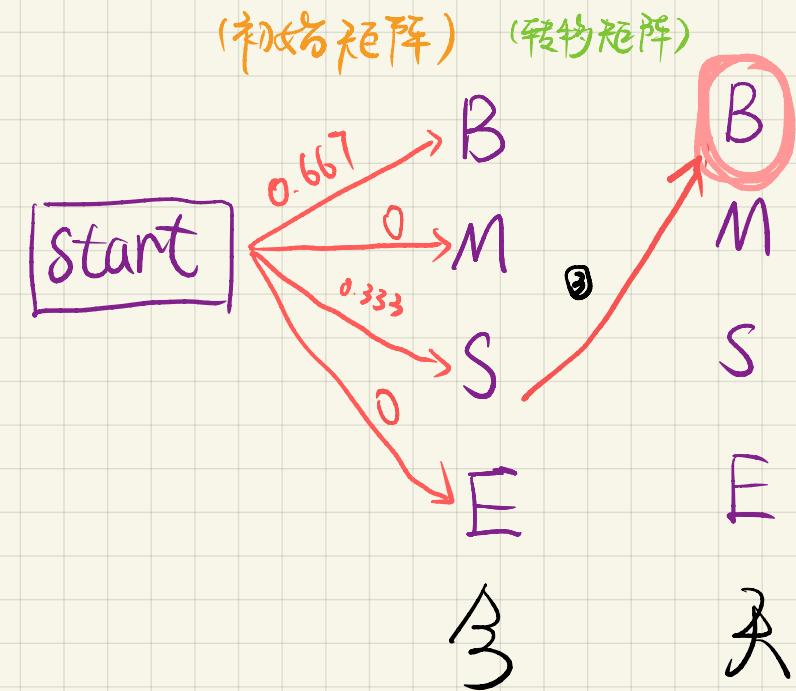
(初始矩阵) (转移矩阵)



③

天

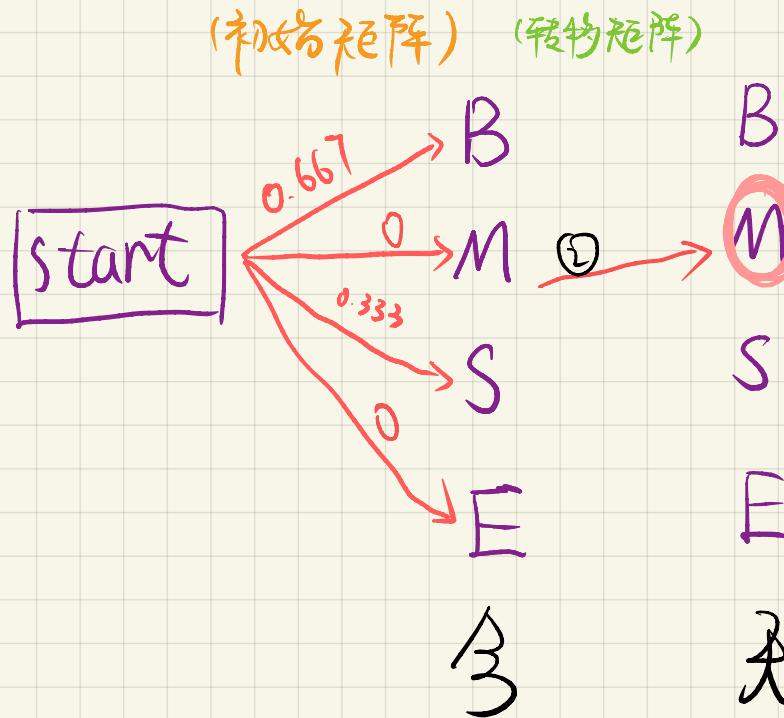
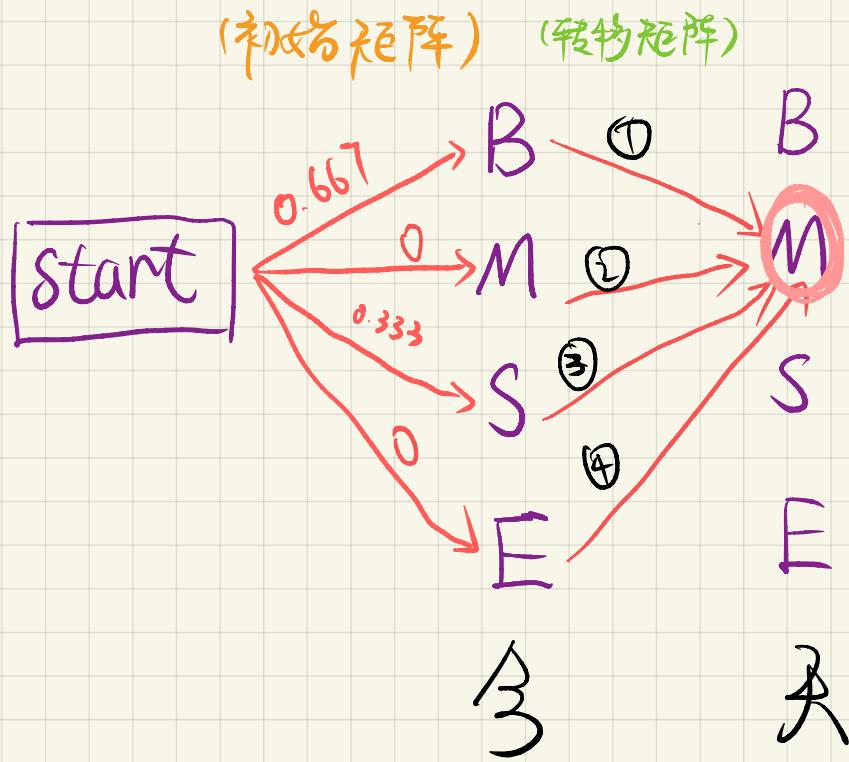
(初始矩阵) (转移矩阵)



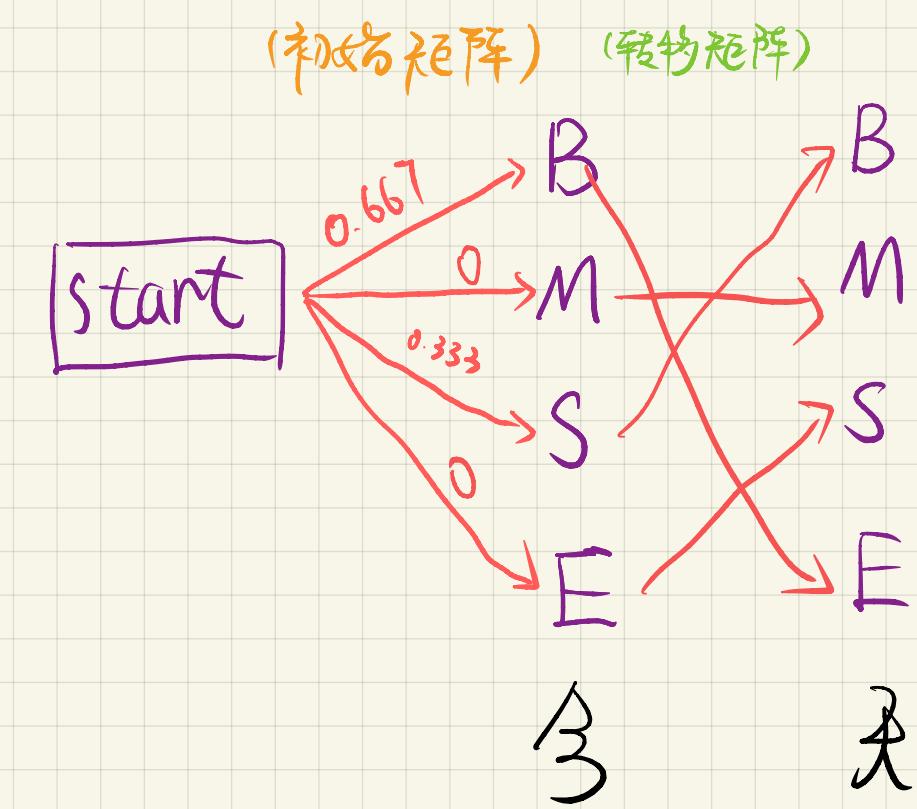
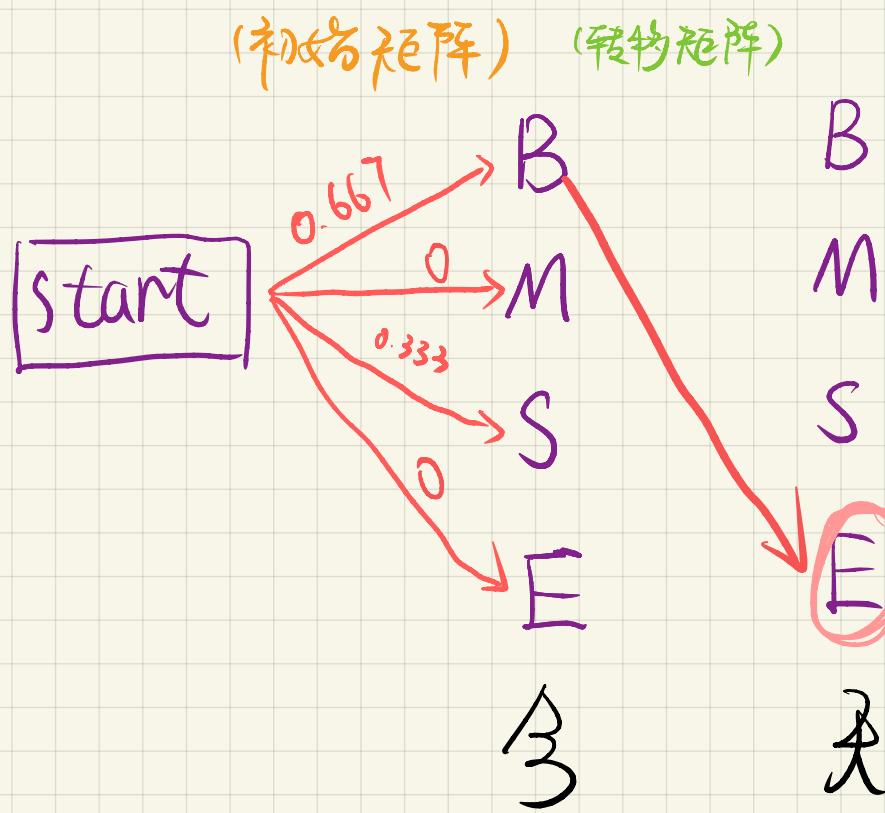
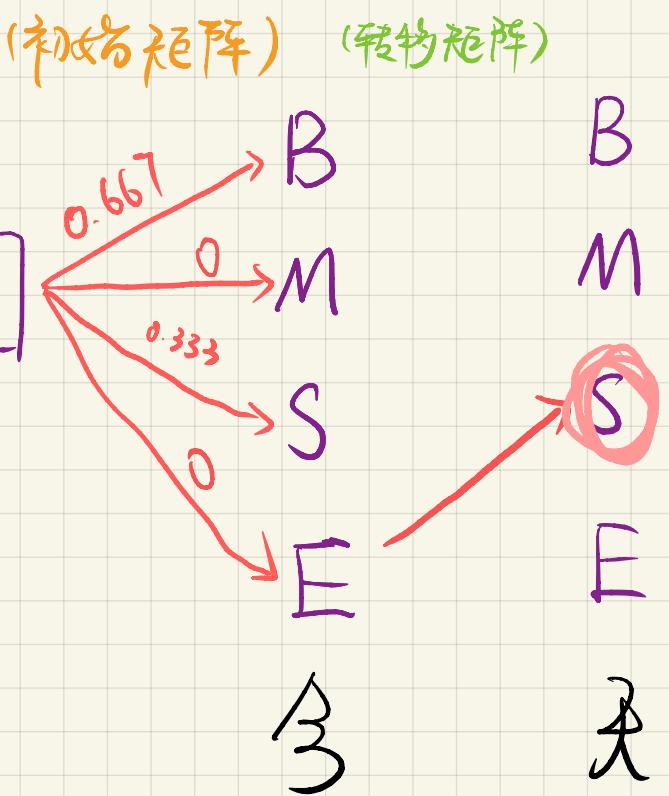
③

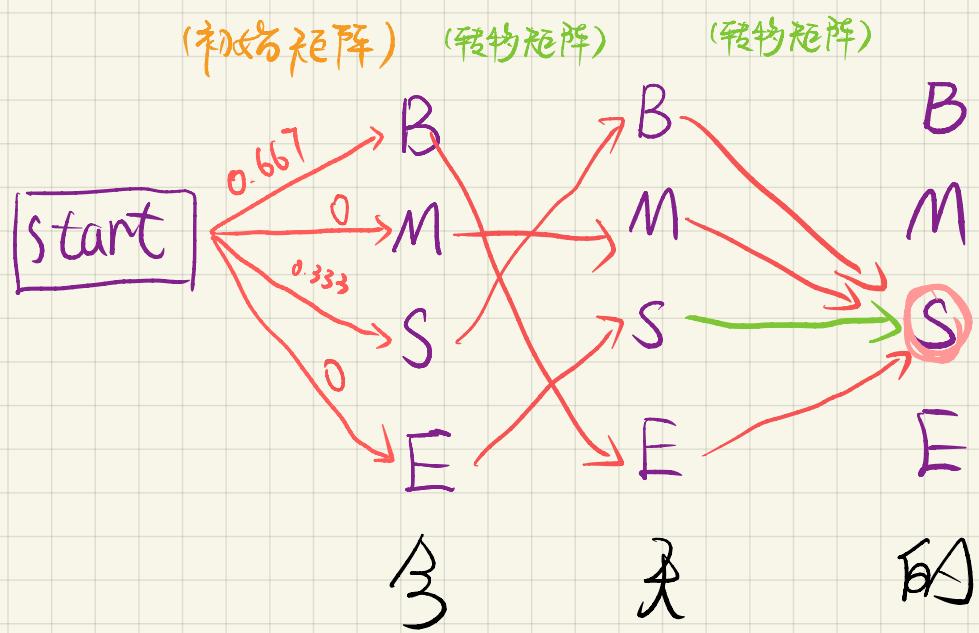
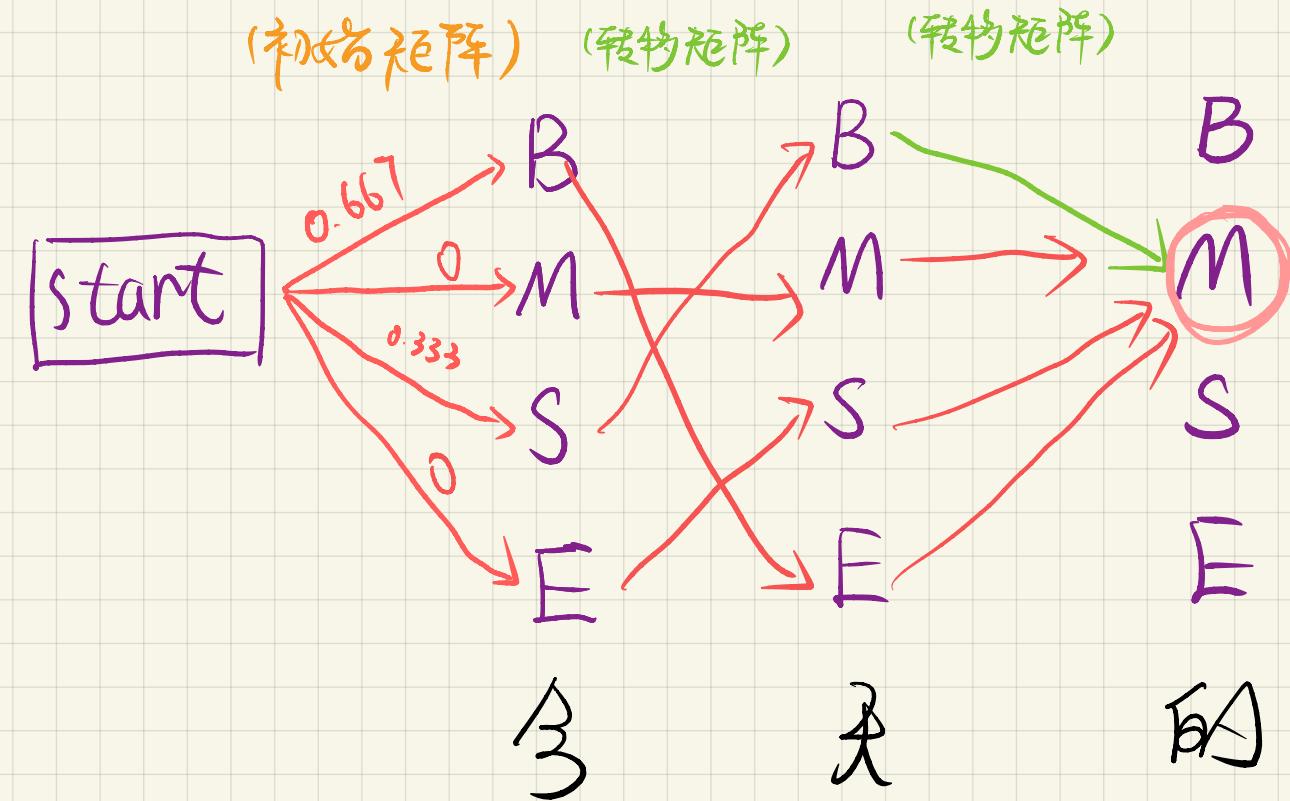
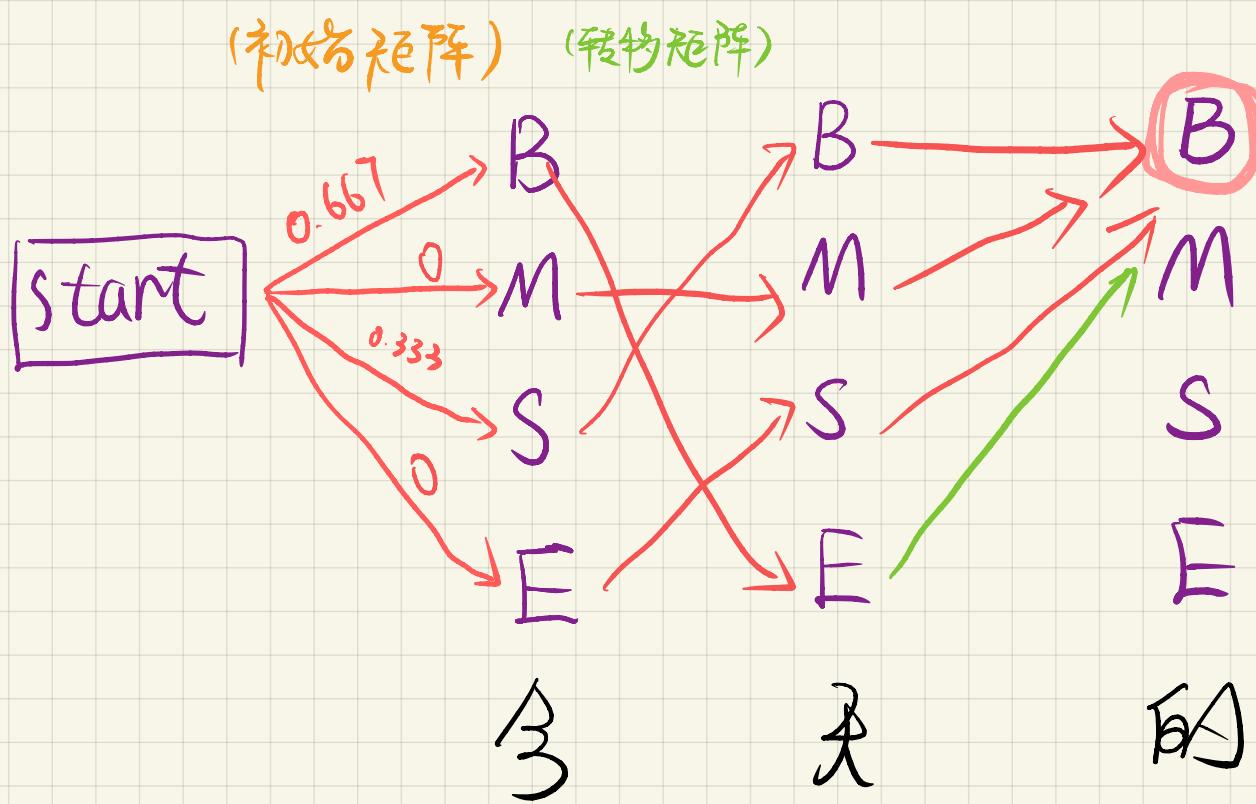
B  
M  
S  
E  
天

(若想到达 B, 有上(1~4) 4条路  
径, 其中必有一条最  
优路径, 假设为 ③,  
其他会被删除)

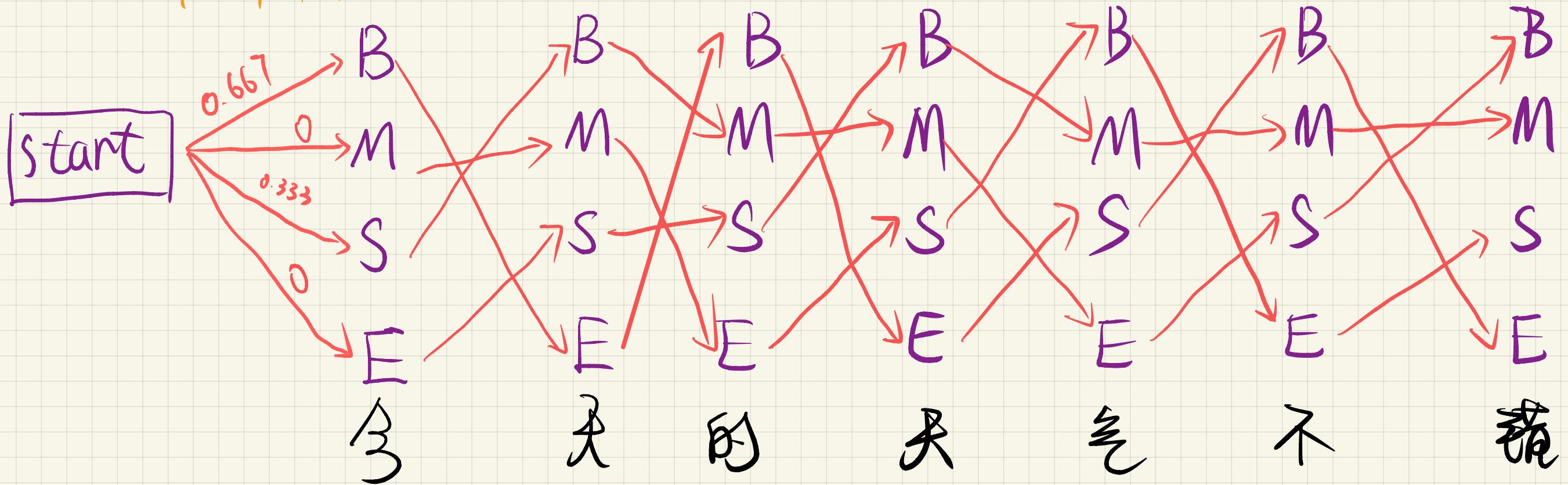


(若想到达  $M$ , 有  
上步 ①~④ + 来路  
路, 其中必有一条最  
优路径, 假设为 ②  
其他会被删除)





(初始矩阵) (转移矩阵)



根据上述方法，再到这一十字都只会有4条路径，在4条路径中，这样最优的，则可得到状态序列  
全局结束

算法

## 如何通俗地讲解 viterbi 算法？

关注问题

写回答

邀请回答

好问题 16

2 条评论

分享

...

43 个回答

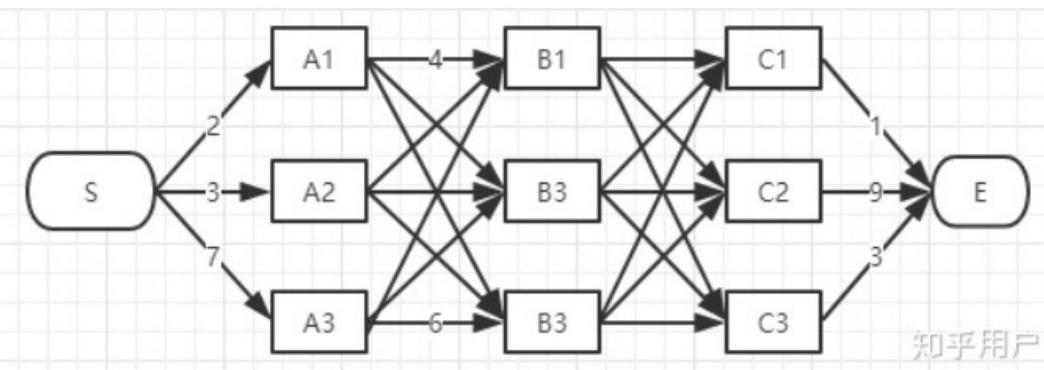
默认排序



知乎用户

1,666 人赞同了该回答

这篇回答你绝对看得懂！如下图，假如你从 S 和 E 之间找一条最短的路径，除了遍历完所有路径，还有什么更好的方法？



知乎用户

viterbi 维特比算法解决的是篱笆型的图的最短路径问题，图的节点按列组织，每列的节点数量可以不一样，每一列的节点只能和相邻列的节点相连，不能跨列相连，节点之间有着不同的距离，距离的值就不在图上——标注出来了，大家自行脑补