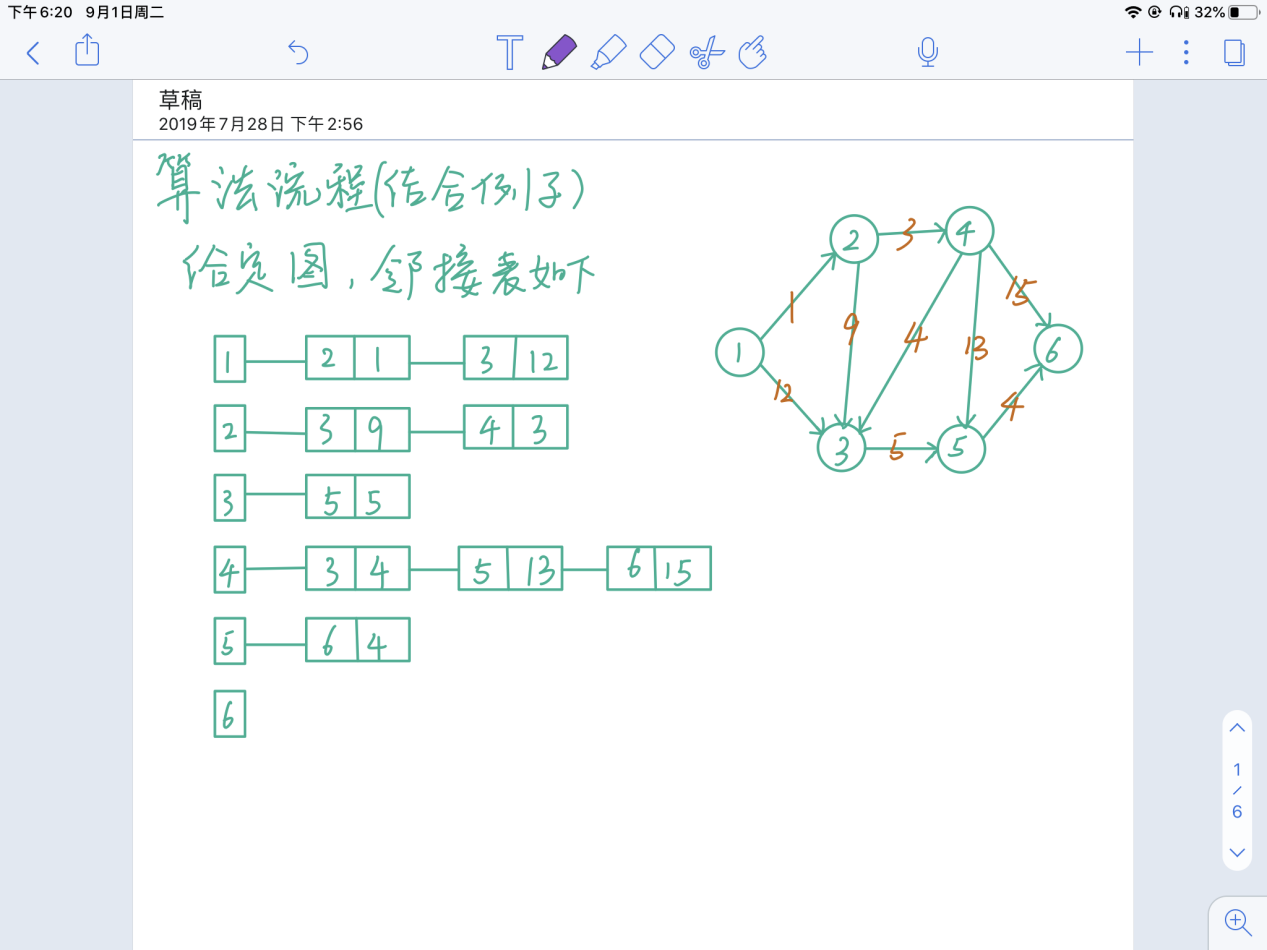
# dijkstra算法



算法伪代码为：

1. 将整个集合分为两个X,Y，初始化的时候X{1}，Y{2，3，4，5，6} d(1)=0，d函数用来记录点到初始点1的距离
2. for y:2 to n

if y相邻于1 then d(y)=len(1,y)，len函数用来返回边的长度

else d(y)=无穷

end for

1. for j:1 to n-1

从Y中取出点y，使得函数值d(y)最小，这里可以使用堆优化

将y添加入X

将y从Y中去掉

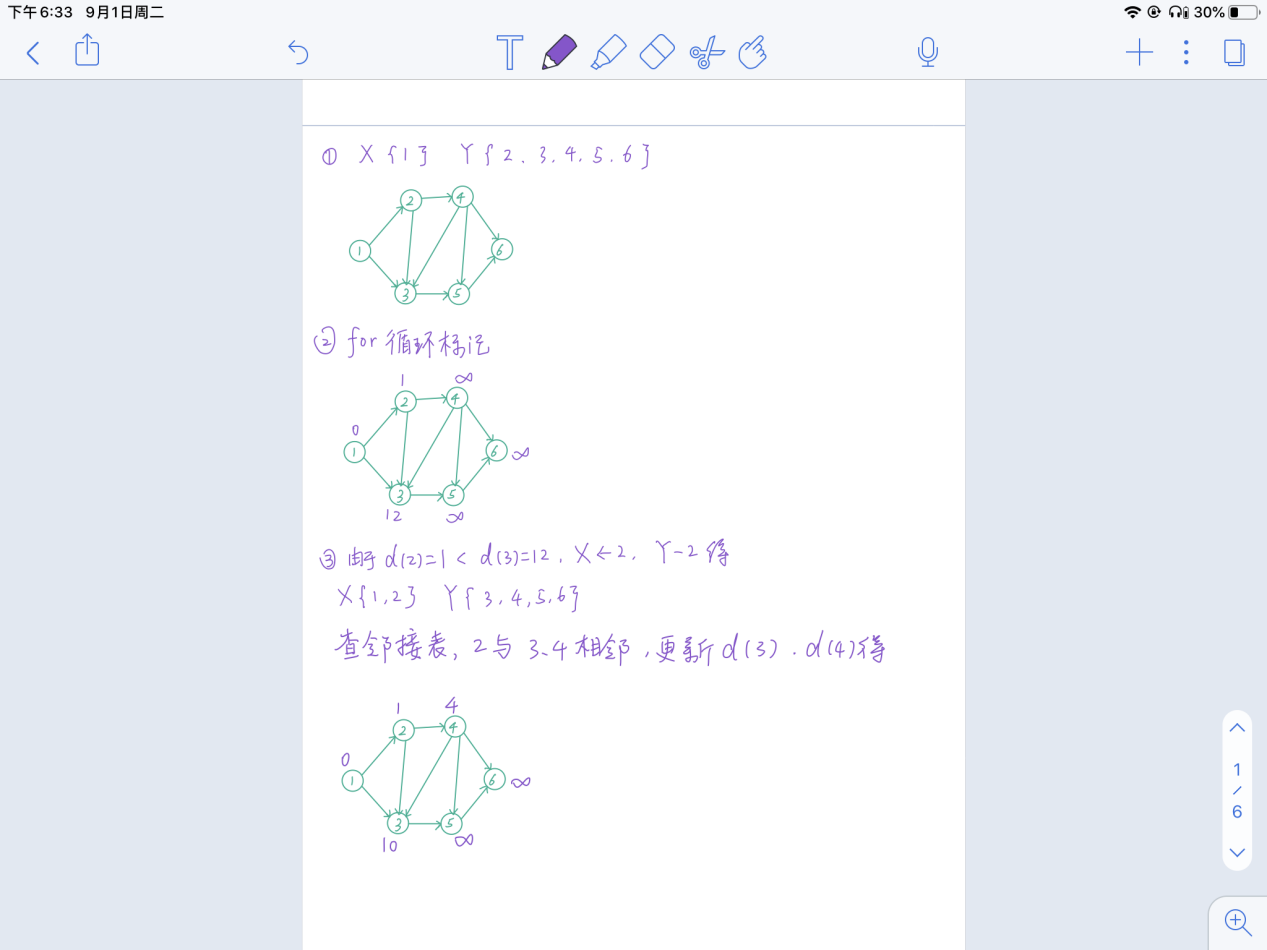
for 每条边(y,w)

if w属于Y and d(y)+len(y,w)<d(w) then d(w)=d(y)+len(y,w)

end for

end for

一部分算法流程如下：



实际使用的算法伪代码

1. 创建用于堆的数组（边数组）
2. 初始化创建的数组，权值全部设为无穷
3. 把起点的内容加进去，起点的权重为0
4. 把邻接于起点的点的内容更新（更新堆里面的数据，由于初始化的时候已经把堆数组建成了，因此所有顶点都已经入堆，出堆的时候，维护堆的结构只需要用一个整数记录堆数组的大小就行了，也就是n），权重为与原点相连的边的权重
5. 从堆中取出堆顶的点y，访问它的邻点w，d(y)+len(y,w)<d(w) then d(w)=d(y)+len(y,w)，也就是更新点w到原点的距离
6. 最后堆数组逆序就是从原点到终点的最短路径，而且堆中每个元素的权重就是原点到这个点的最短距离