小世界现象

姜少峰

北京大学前沿计算研究中心
Center on Frontiers of Computing Studies, Peking University

六度分隔：Stanley Milgram的研究

- 1960年代， Stanley Milgram做了一个送信的社会实验
  - 随机选择一些“起始人”，要求他们向某个陌生“目标”转发一封信
  - 知道“目标”的基本信息，比如姓名、地址、职业等，但不能直接寄往地址
  - 需要通过自己的熟人经过中转尽快把信送到
  - 1/3的信件经过平均6次转发到达目标

说明的问题

- 说明网络的直径比较小，包含丰富的短路径
- 图的直径：图中任何两节点间的最短路距离的最大值

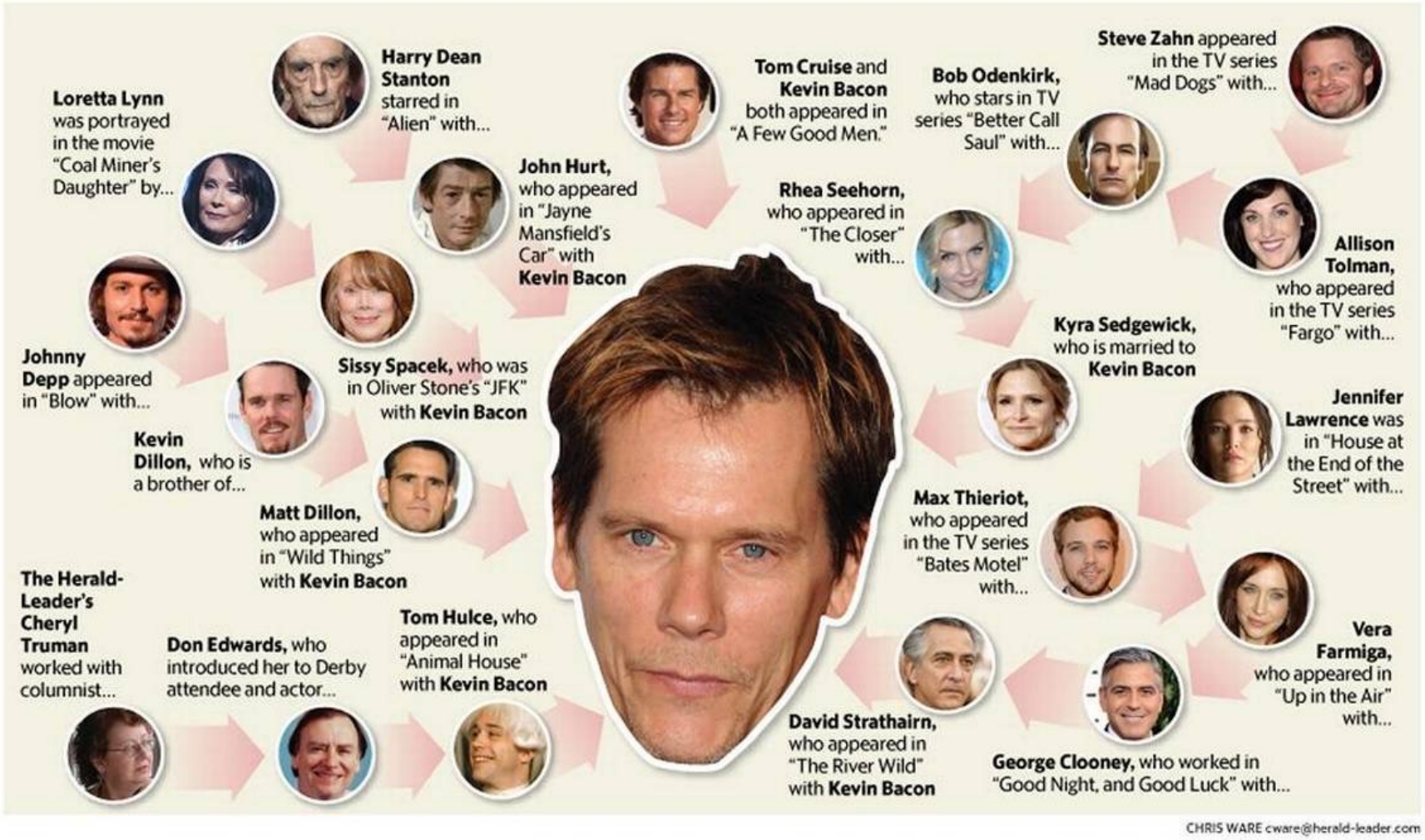
$$\mathrm{diam}(G=(V,E))=\max_{u,v\in V} d_G(u,v)$$

实际网络的直径

http://snap.stanford.edu/data/index.html

Bacon number

https://oracleofbacon.org/



Erdős number

https://www.csauthors.net/distance/paul-erdos/shaofeng-h-c-jiang

估算图的直径

- 如何估算直径?
  - 精确计算很难避免$O(n^2)$
- 2-近似可以线性时间计算
- 图上最短路距离满足三角形不等式（triangle inequality）

$$\forall x, z, y \in V, \qquad d_G(x, y) + d_G(y, z) \geq d_G(x, z)$$

  - 证明：利用最短路最优性的反证法
- 2-近似算法：任取一个点u，找到u的最远点v，报告$d_G(u, v)$
  - 如何线性时间找最远?
  - 为什么是2-近似?

实验反映的另一个事实：短视搜索

- Global vs local：与去邮局寄信的区别？

- 短视搜索：
  - 每个点只知道：
    - 自己邻居的位置
    - 目标的位置
    - 没有全局地图！
  - 从u开始，给定v，每轮只能沿着某条边到达某个邻居，送信到v
  - 需要多少轮可以到达v？

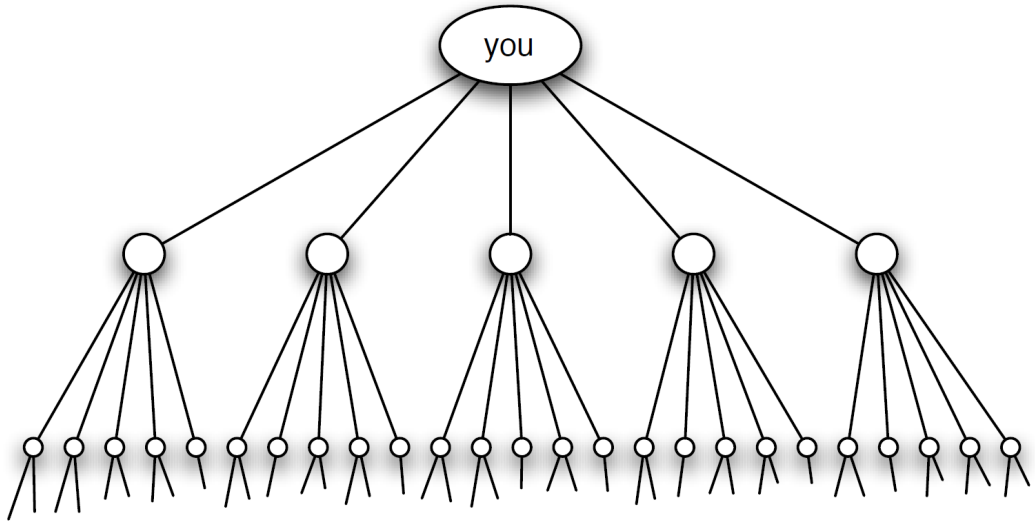
小世界现象的建模

什么是建模?

- 找到一种理论，来解释已有的现象
  - 为什么需要模型：理解现象的本质成因；定量/定性研究成为可能；预测
- 模型都是可证伪的
  - 证伪：在某个数据集/实验上测试发现理论不符合观测
- 自然科学的核心是这种建模-证伪的循环，来不断深化对现象的认识
- 这种建模的方法同样适合于研究社会网络

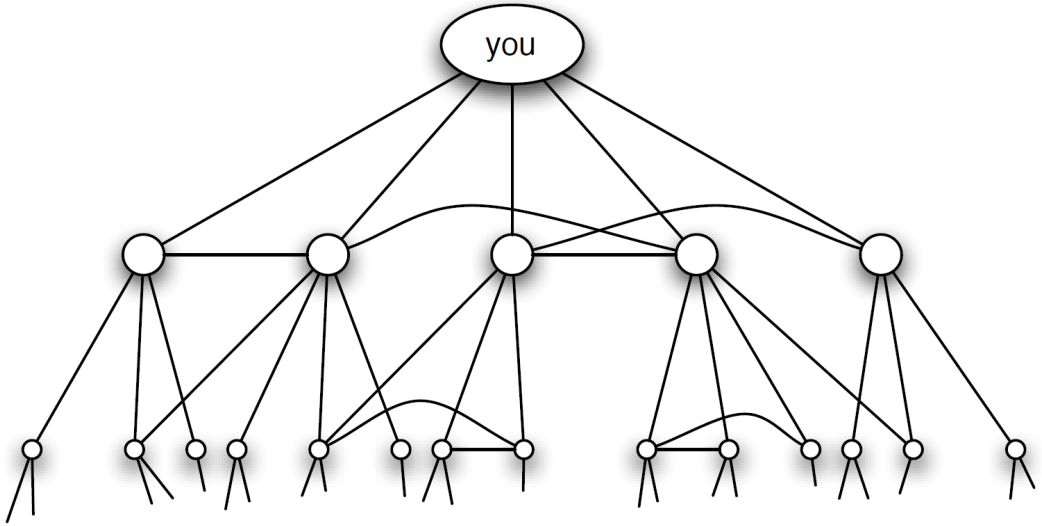
一个简单模型

- 一个简单、自然的模型：
  - 假设每个人都认识100个朋友
  - 通过自己的朋友，可以到达100 x 100 = 10000人；朋友的朋友可达100万人...



模型的问题?

- 模型“不像”社交网络
  - 朋友的朋友能有1万人吗?
- 问题：三元闭包的存在，极大地限制了两步可以到达的朋友数量
  - 这也是六度分隔反直觉的地方：局部大量重叠聚集，无明显通向远方的路径



如何改进模型？什么是好的模型？

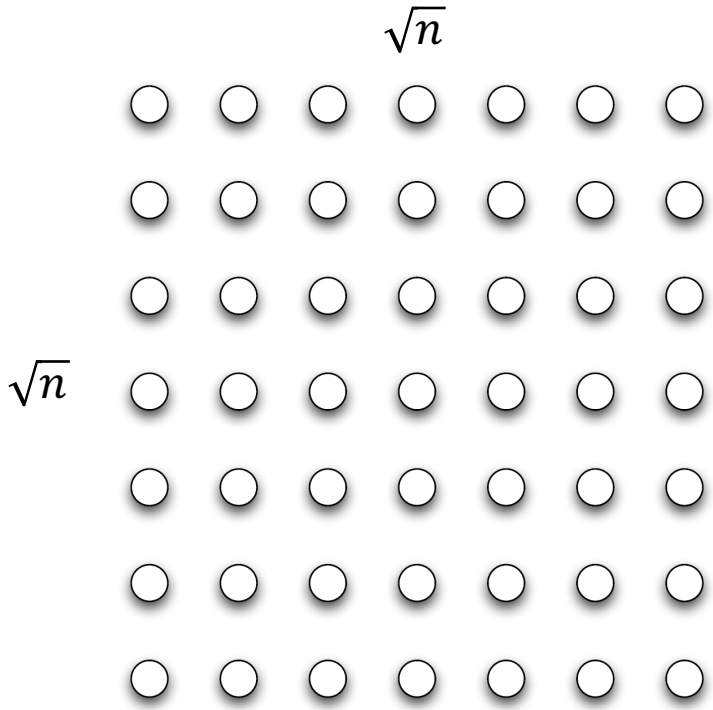
- 奥卡姆剃刀：在能解释现象的若干模型中，选取尽量简单/假设少的
  - 对于“可证伪性”尤为重要：总可以找到复杂理论囊括所有观测，但无意义
    - 例如：给出数列前几项猜通项公式，m项总可以有O(m)次多项式的通项公式
    - 避开“过拟合”，“特例假设”
    - 越复杂的理论越难以证伪
  - 太过复杂可能会引入一些不必要的细节，反而掩盖/阻碍了对真正本质的认识

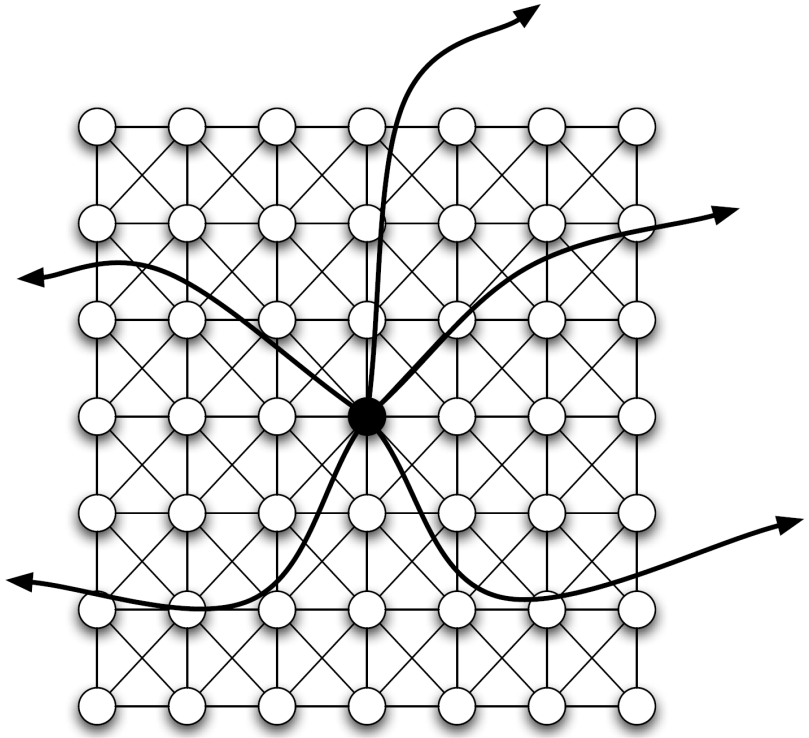
小世界模型应该具备的性质

- 满足三元闭包
- 有丰富短路径/直径小
- 同时具备其他社会网络的属性
  - 例如弱连接（“桥接”远处节点）
  - 同质性
- 尽量简洁

Watts-Strogatz模型 (Watts and Strogatz, 1998)

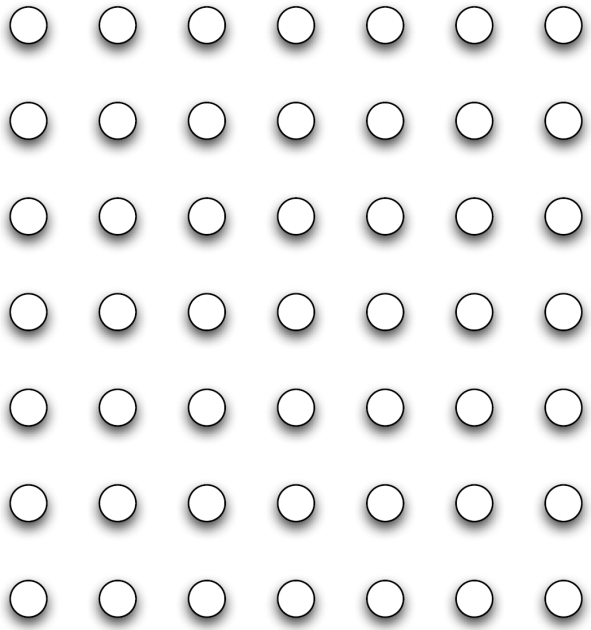
- WS随机网络：假定人在二维网格中，上下左右相邻为一个网格步
  - 每个节点创建两种连接（设r和k是预先固定的常数）
    - 同质性连接：连接所有r网格步能到达的所有人
    - 弱关系连接：从整个二维网格以均匀分布独立随机地选取k个点进行连接

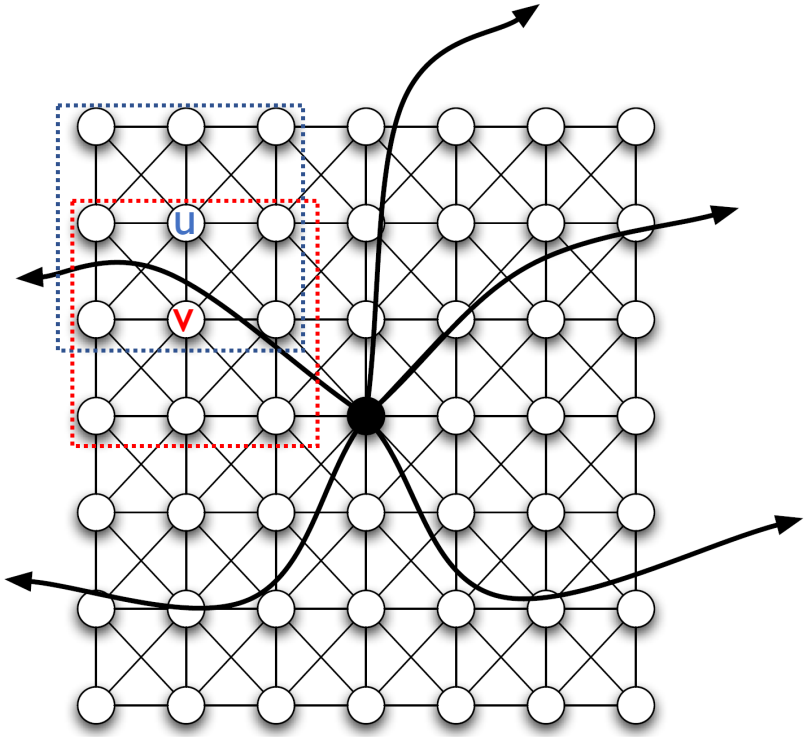




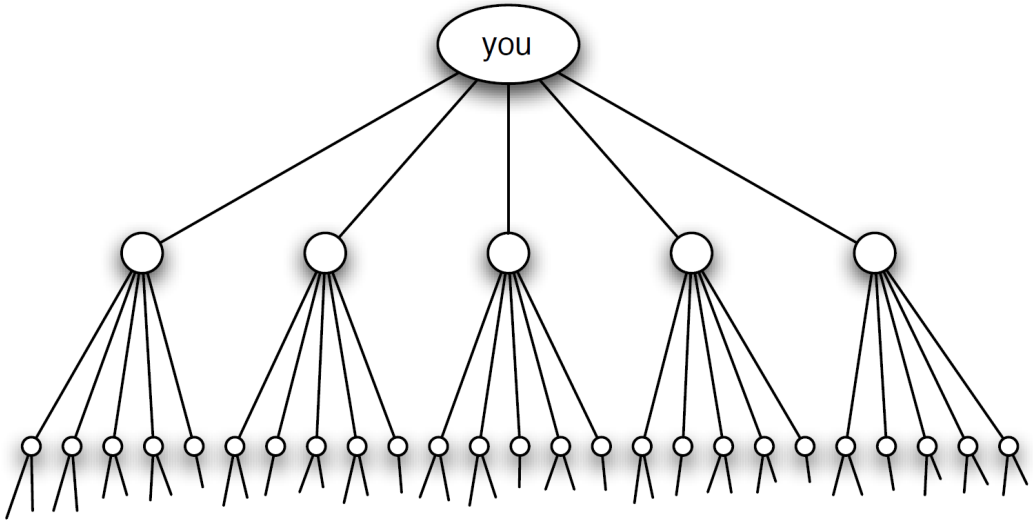
Watts-Strogatz模型的典型行为

- 三元闭包
  - 只要两人距离r以内，就能有很大的重叠，很多三角形



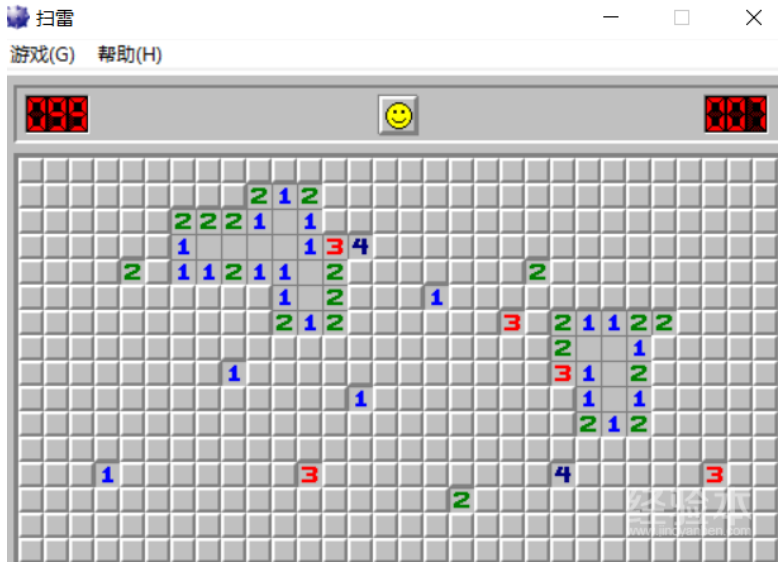


- 任意两点有短路径：一种随机的、类似BFS的搜索过程
- 结论：直径大概率O(poly log n)
- 考察：任取一点u，考虑u经过多少边可以访问到所有点
- 现象：起初，会出现完全二叉树类似的探索过程



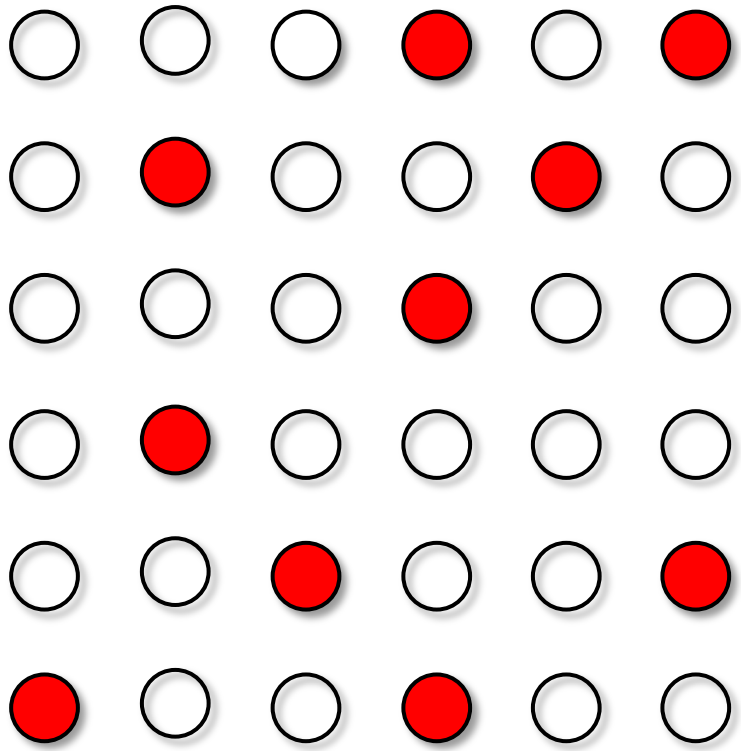
第一阶段：完全二叉树/BFS式遍历

- 初始：u沿直接连接走若干步，访问到的邻居各自访问k = l个随机点
- 每轮从上轮新达点沿直接连接走若干步，然后新邻居连到各自随机点
- 若还有0.5n的人没访问过，随机边终点大概率落在没有访问过的区域
  - 这样每轮让访问到的人数乘以一个常数c，第i轮结束会累计到达大约c^i个人
  - 到0.5n人已经访问的时候，仅需要O(log n)轮



第二阶段：走同质性连接

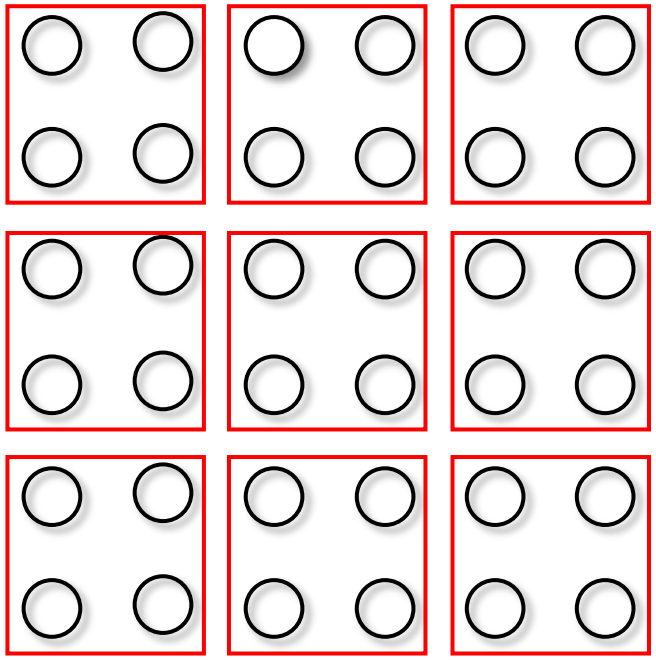
- 最后如果已经访问到了0.5n的人，剩下的0.5n?
  - 大概率已访问的点均匀分布在格点上，其他点到他们的距离大概率O(log n)



Coupon collection

Coupon collection: 有n种均匀盲盒, 则独立开O(n log n)个能以$1 - 1/n$概率n种全集齐

- 对每个i，O(n log n)次后i没收集的概率是$\left(1 - \frac{1}{n}\right)^{O(n \log n)} \leq \frac{1}{poly(n)}$
- 存在某个i在O(n log n)次后还没集齐的概率至多n倍的上述概率，也就是1 / n



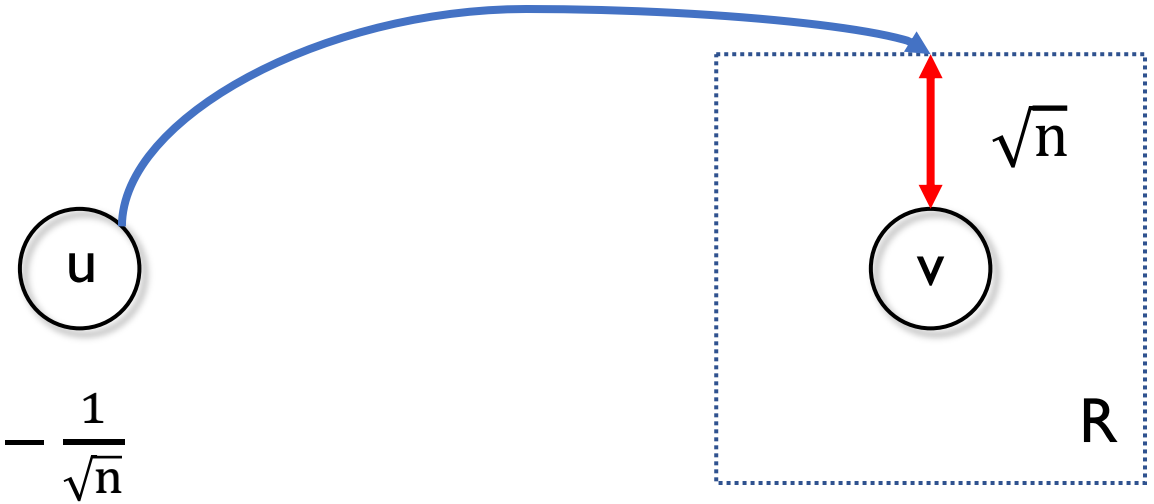
将格点均匀分成O(n / log n)个正方形
然后用coupon collection

路径短能说明搜索也容易吗?

- 这个长O(poly log n)的路径需发动所有邻居搜索才能找到，类似于BFS
- 因此，WS模型在这种分布式/短视的情况下很难找到短路
- 然而在Stanley Milgram实验中，每次只能把信托付给一个朋友转发

WS模型短视搜索需要多少步?

- 假设u, v足够远，距离$\Omega(n)$
- 考虑v附近$\sqrt{n}$ 距离的区域R
  - 假设随机边数k = 1
  - 假定u到R之前只走随机边
  - 每次走后还在R外的概率是$1-\frac{\sqrt{n}}{n}=1-\frac{1}{\sqrt{n}}$
  - 经过$\sqrt{n}$步后还没进入R的概率是常数



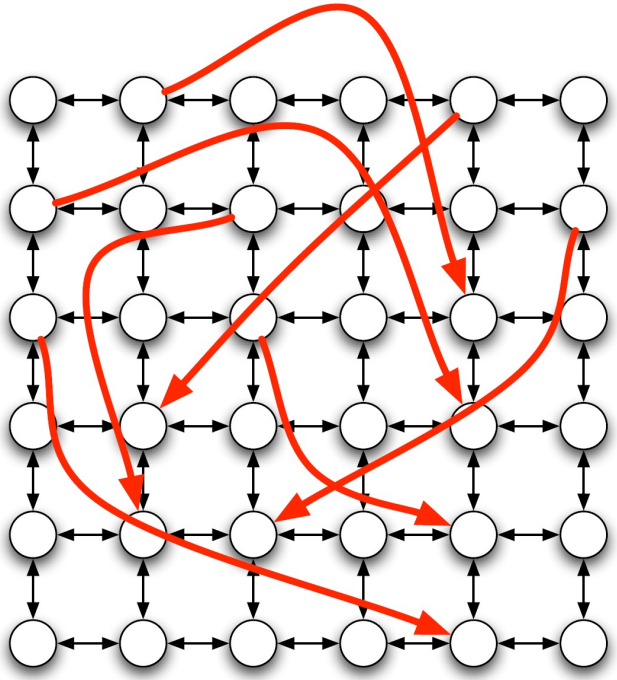
- 因此，总共需要走至少$\sqrt{n}$步，即$\Omega(\sqrt{n})$步才能到v!

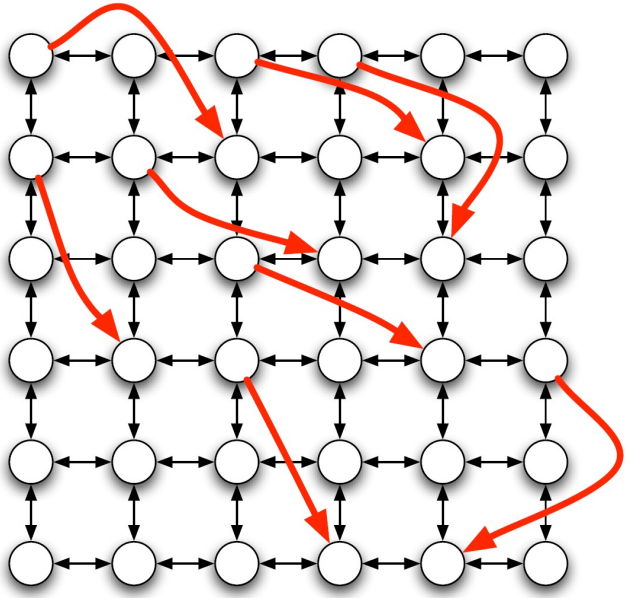
- 模型问题所在：WS的弱连接“太随机”，没有任何方向性、结构

WS的扩展：支持短视搜索(Kleinberg, 2000)

- 同质性连接不变：每人依然连接自己的r网络步以内的邻居
- 弱关系连接需要更“有结构”
  - 对于点u，建立一条到点m的概率正比于$d(u,m)^{-q}$
  - 这里$q \geq 0$是一个参数

q较小

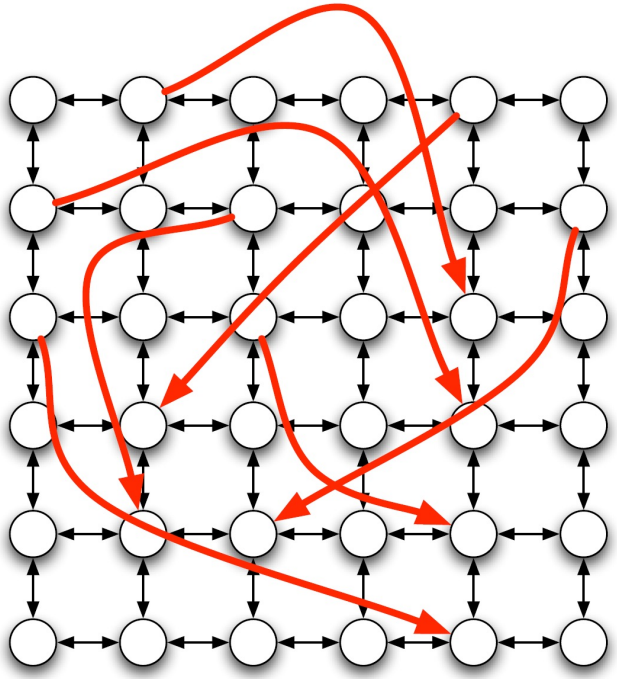


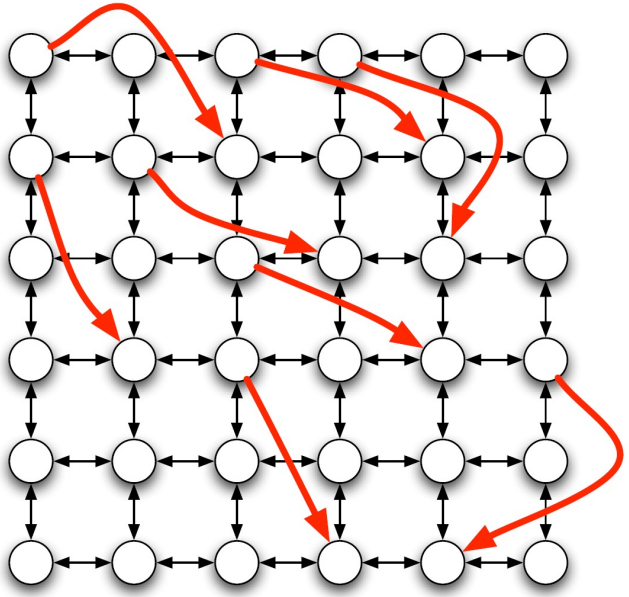


q较大

- 为何与距离成负相关?
  - 足够远的点的连接需要概率低一些，否则“太随机”，会跨步太大
  - 另外，认识距离自己太远的人本身也应该是小概率事件
  - 同时，也不能概率都集中于近的点，否则不能形成有效跨步，最短路会太长
  - 因此，需要一个“平缓”的根据距离负相关的概率分布

q较小

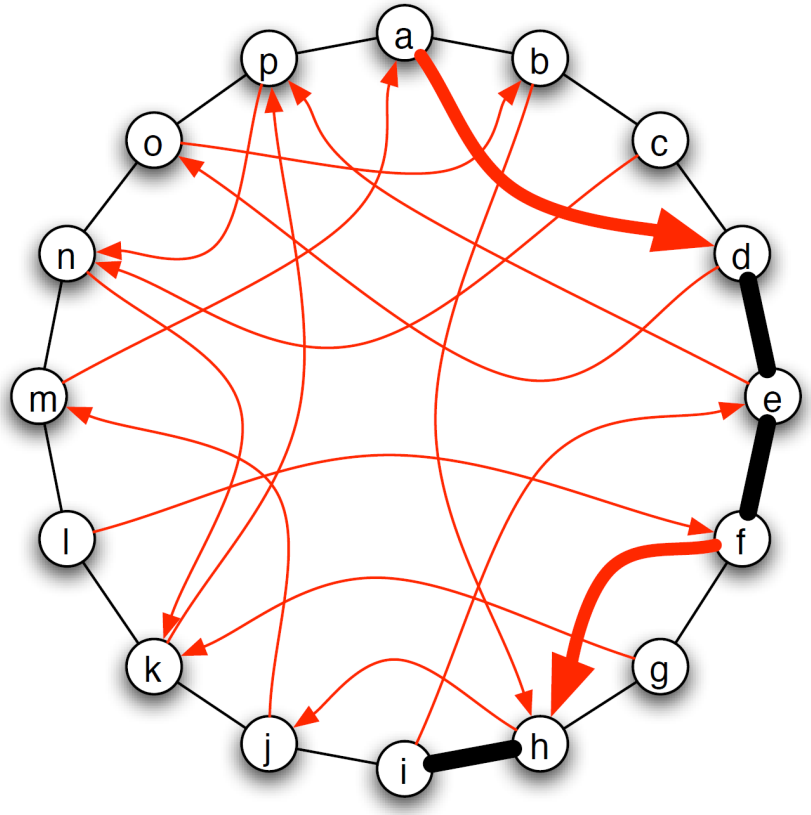




q较大

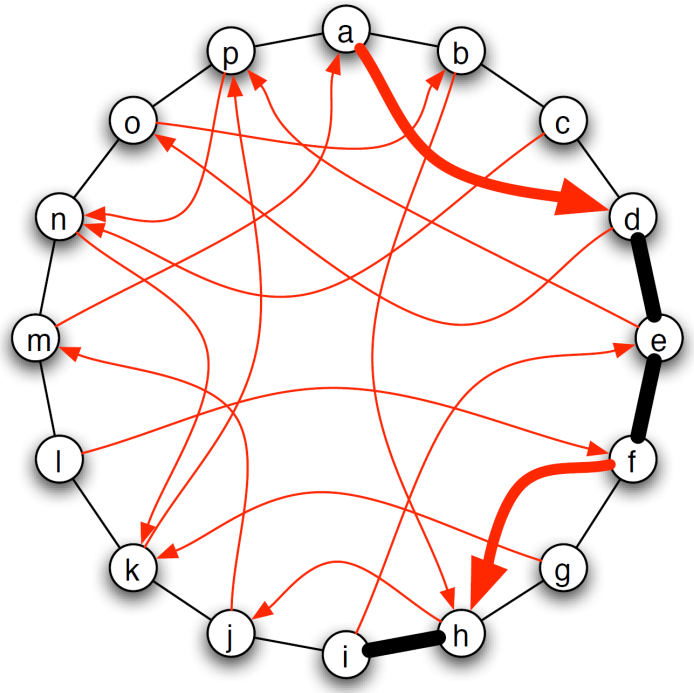
1D ring的WS模型的短视搜索

- 1D ring：n个点排成一个环，每个点连接左右相邻的两点
- 1D ring的WS模型：
  - 每个点u连出一条有向随机边，其中连v的概率正比于$d(u,v)^{-1}$



基本概率分析

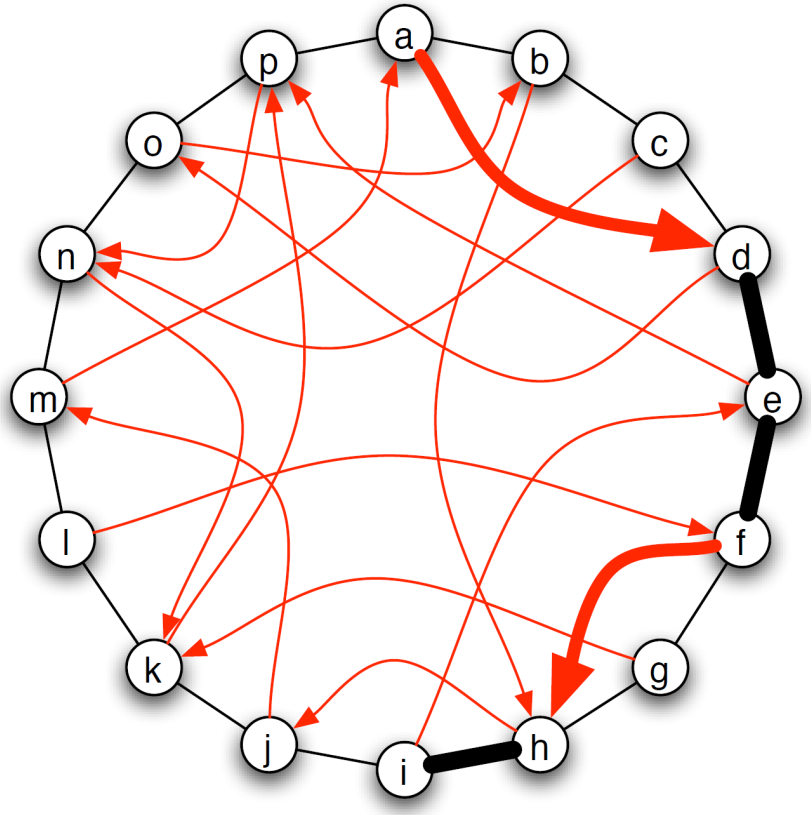
- 对于某个点u，随机边连接到v的概率是多少?
  - 根据定义，对于每个点v'，u连到v'的概率正比于$d(u,v')^{-1}$
  - $\Pr[u\text{的随机边连到v}] = \frac{d(u,v)^{-1}}{\sum_{v'} d(u,v')^{-1}} = \frac{d(u,v)^{-1}}{2\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{(2/n)}\right)} = \frac{d(u,v)^{-1}}{2H_{n/2}}$
  - $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n$



短视搜索策略：贪心

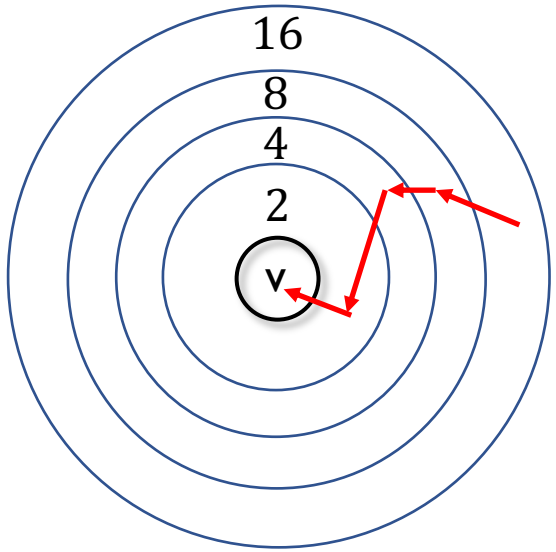
- 给定起点u终点v，我们采用贪心的短视搜索策略
  - u从三条连边中选取使到v距离减小最多的一条
- 例如：
  - 起点a终点e，那么a应该选取到d的随机边
  - 起点a终点h，那么应该走a->d->e->f->h
    - 这不是最短路；最短路：a->b->h
- 我们要证明：

对于任意的u和v，贪心策期望在$O(\log^2 n)$步内结束



证明策略

- 固定一对u和v
- 注意到：算法执行中，到v的距离每次都在单调缩小
- 想法：
  - 将到v的距离按照2的方幂进行划分，分为log n组
  - 从$i+1$组走到i组会将距离缩小大约一半
  - 论证从$i+1$组到i组花费的期望步数是O(log n)
  - 总共就是$O(\log^2 n)$

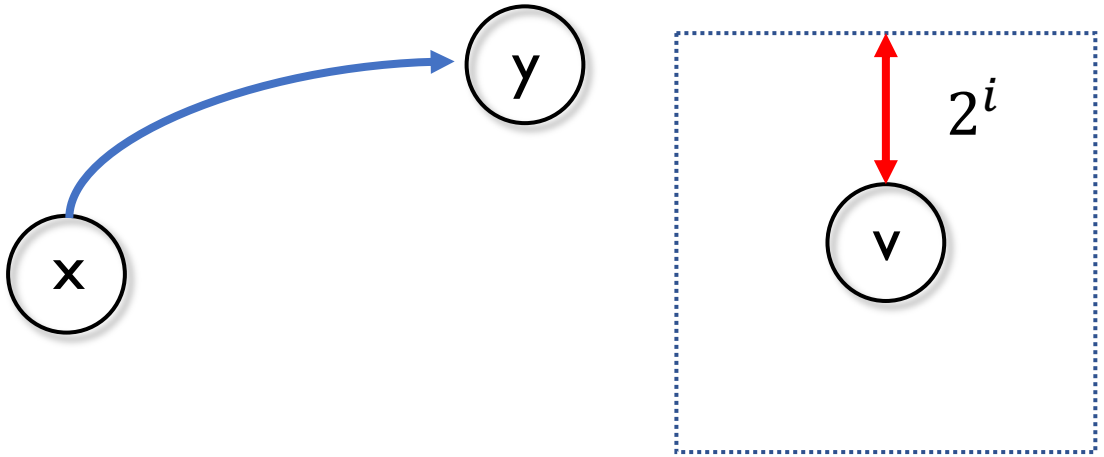


为何不按照之前讨论的u -> u'…，而是直接从v的视角划分?

期望分析

- 设：随机变量X_i代表距离v在$[2^i, 2^{i+1})$所花费的步数，设X是总步数
- $X = X_1 + \cdots + X_{\log n}$ $\Rightarrow$ $\mathbb{E}[X] = \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_{\log n}]$
- 下面，固定一个i，分析X_i的行为
- 设$P_j := \{u \mid d(u,v) \in [2^j, 2^{j+1})\}$代表到v距离在$2^j$附近的点

$\Pr[X_i > t] \leq \Pr[$有t个点$x \in P_i$使得x的随机边(x,y)满足$d(y,v) \geq 2^i]$

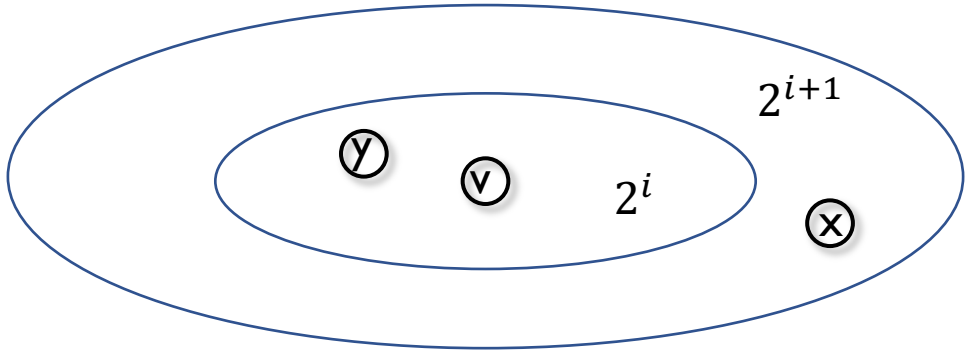


期望分析

- $\Pr[X_i > t] \le \Pr[$有t个点$x \in P_i$使得x的随机边(x,y)满足$d(y,v) \ge 2^i]$
- 一个点：
  - $\Pr[x \in P_i$使得x的随机边(x,y)满足$d(y,v) \ge 2^i]$
  - $= 1 - \Pr[x \in P_i$使得x的随机边(x,y)满足$d(y,v) < 2^i]$
  - $\Pr[x \in P_i$使得x的随机边(x,y)满足$d(y,v) < 2^i]$
  - $\ge \sum_{k=2^i}^{2^{i+1}} \frac{1}{kH_{n/2}} = \frac{H_{2^{i+1}} - H_{2^i}}{H_{n/2}} \approx \frac{1}{H_{n/2}}$
  - $\Pr[X_i > t] \le \left(1 - \frac{1}{H_{n/2}}\right)^t$

$$\Pr[\text{x的随机边连到}y] = \frac{d(x,y)^{-1}}{2H_{n/2}}$$

$$H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n$$



期望分析

- 已有：$\Pr[X_i > t] \leq \left(1 - \frac{1}{H_{n/2}}\right)^t$
- 公式：对于非负随机变量Y，$\mathbb{E}[Y] = \sum_{k=0}^{\infty} \Pr[Y \geq k]$

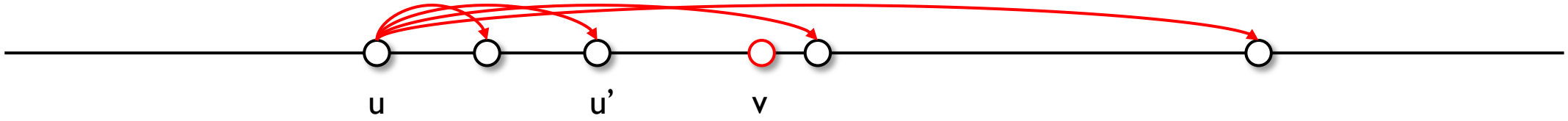
- 应用公式：$\mathbb{E}[X_i] = \sum_{k=0}^{\infty} \Pr[X_i > k] \leq H_{n/2}$
- 得出结论：$\mathbb{E}[X] = \sum_{i=1}^{\log n} \mathbb{E}[X_i] \leq \sum_{i=1}^{\log n} H_{n/2} \leq O(\log^2 n)$

证明中用到的一个重要性质

- 在证明中，有用到一个重要性质：对随机边(u,v)，

$$\Pr[d(u,v) \in [2^i, 2^{i+1}]] = O\left(\frac{1}{\ln n}\right)$$

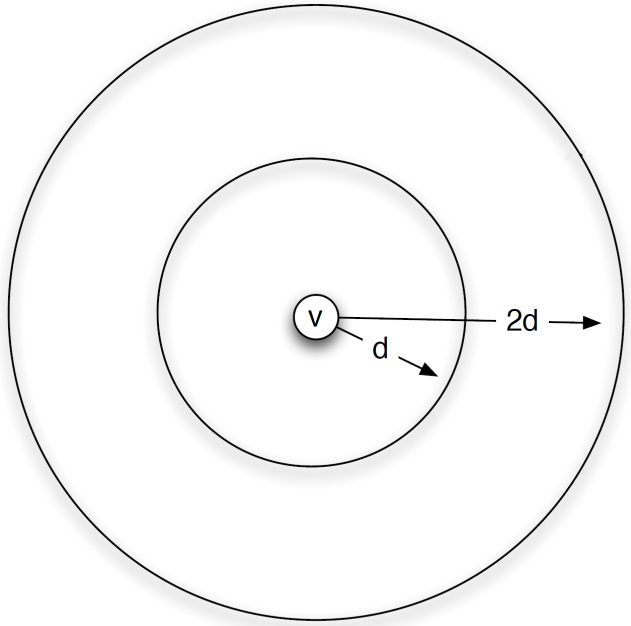
- 即，随机边的长度在每个$[2^i, 2^{i+1}]$范围的概率都是相等的
- 这个性质在证明中的重要作用是?



- 在1D ring上，这个性质依赖于选取q = 1（其他值是不行的）

二维q的选取？

- 我们想找一个q，使得大概是关于O(log n)个分层的均匀分布
- 考虑以u为圆心，半径在r和2r间的圆环
  - 里面含有的点数正比于d^2，如果取q = 2，则与距离平方反比$\frac{1}{d^q} = \frac{1}{d^2}$相抵消
  - 每个层都总体以近似均匀的分布被采样



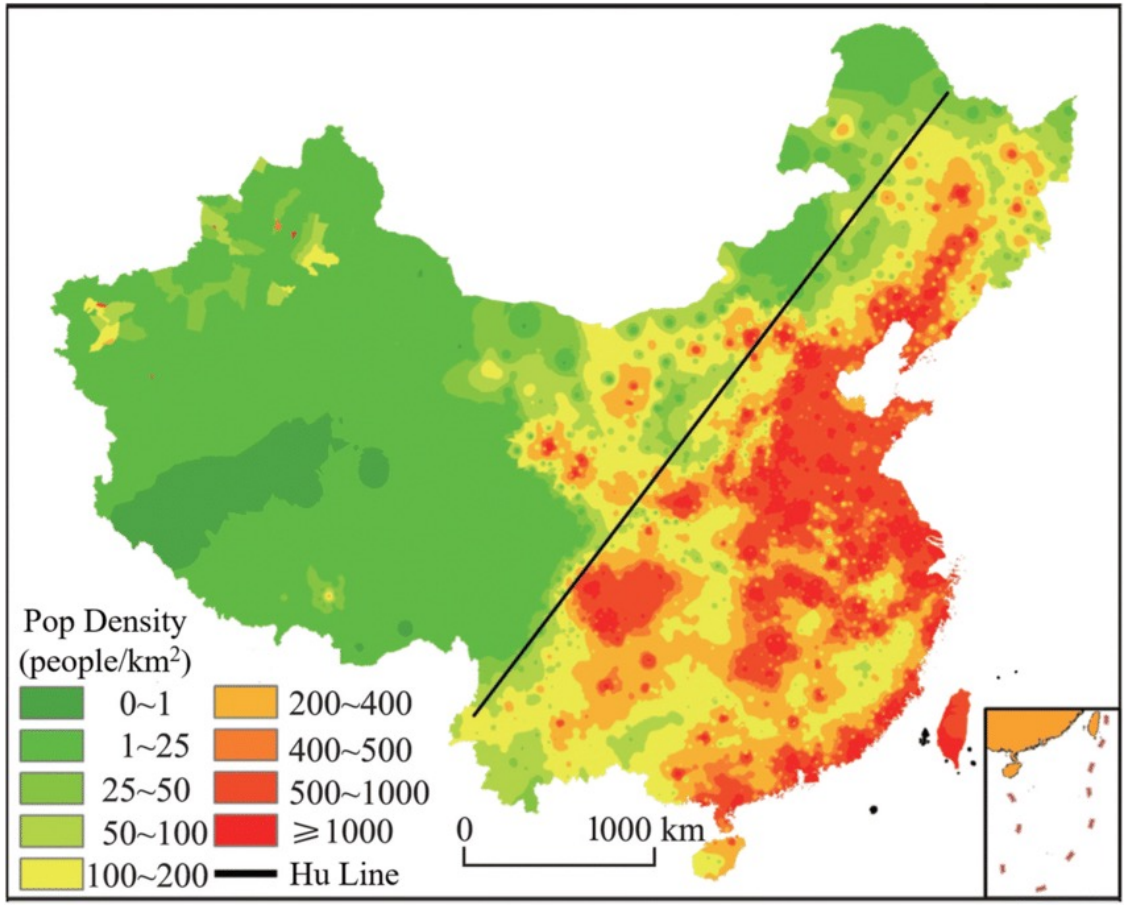
一般地，半径为r的d维球含有r^d个格点

原因：考虑体积

q值在其他情形下的选取

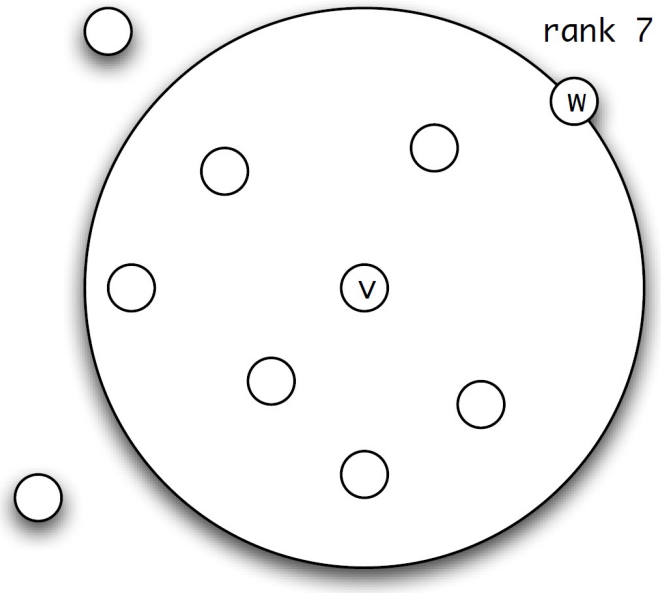
- 有时格点网络建模不再适用，而q = 2是对于格点网络选取的

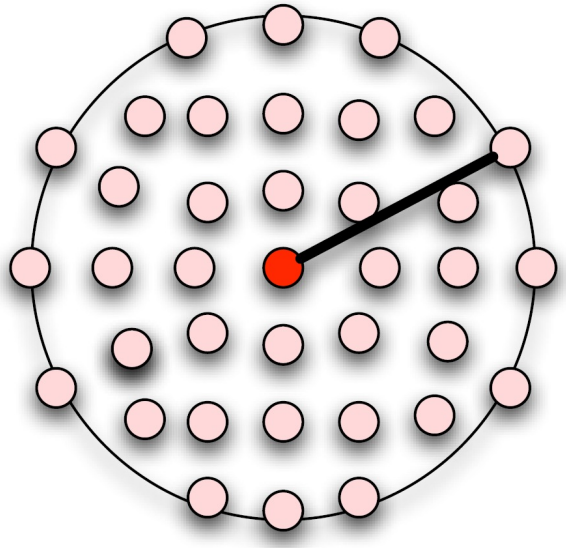
- 人在地理上未必均匀，因此建模成格点不合适
- 此时，可以考虑以距离排名当作新的度量
- 固定节点u，可以定义rank(w)为比w更接近u的点数
- 网络的随机连接规则变成与rank(w)的-q次成正比



rank距离下q怎么取?

- q此时应该等于1
- 还是考虑均匀情况，那么在半径r的圆里面有r^2个点，排名也是r^2，那么距离平方反比就是rank的反比



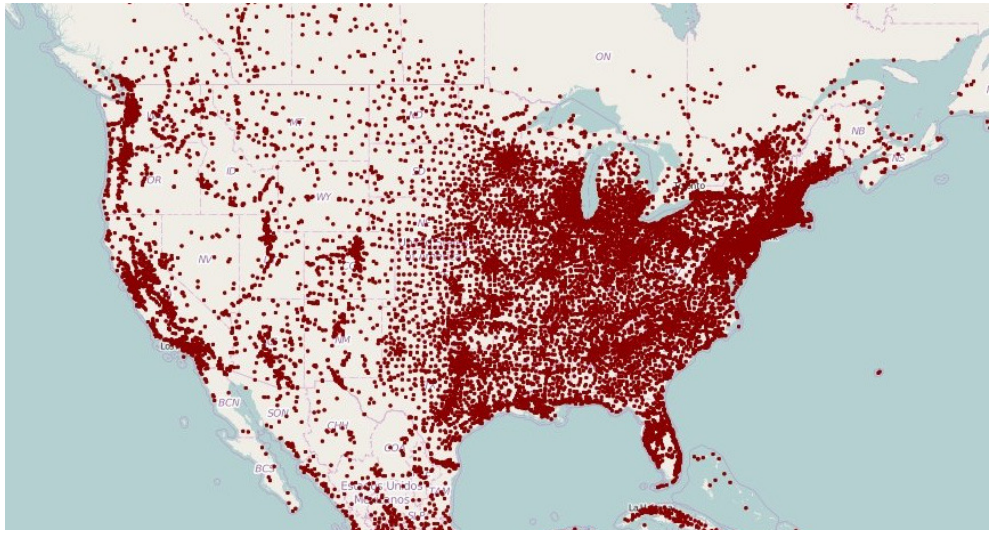


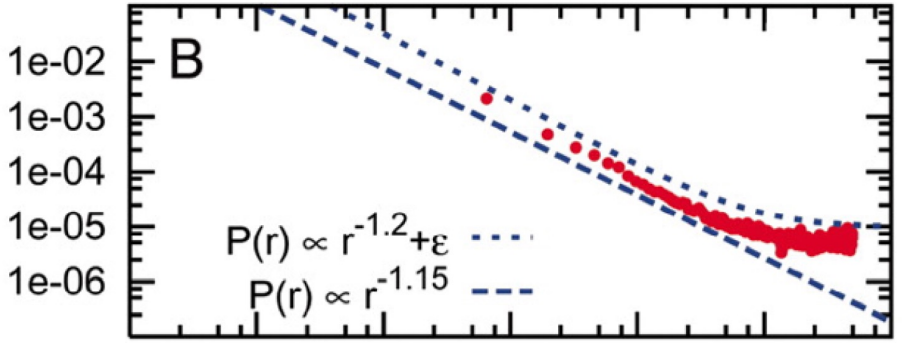
距离/半径是r

rank就对应于r^2

rank距离的q值的实证研究

- 这种改进的WS模型以及q值的选取理论得到了一些数据的支持
- 考虑rank距离
  - 对每个rank r，找出所有点对中，一个点是另一个点rank r的点占所比例/概率
  - 画出这个比例f关于r的函数
  - 为得出使得f $= O(r^{-q})$的q，应该画log f关于r的曲线





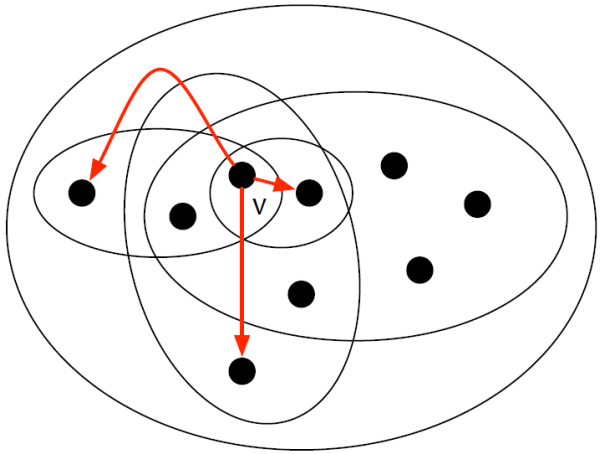
结论：确实按照正比于大约r^{-1}概率进行连接

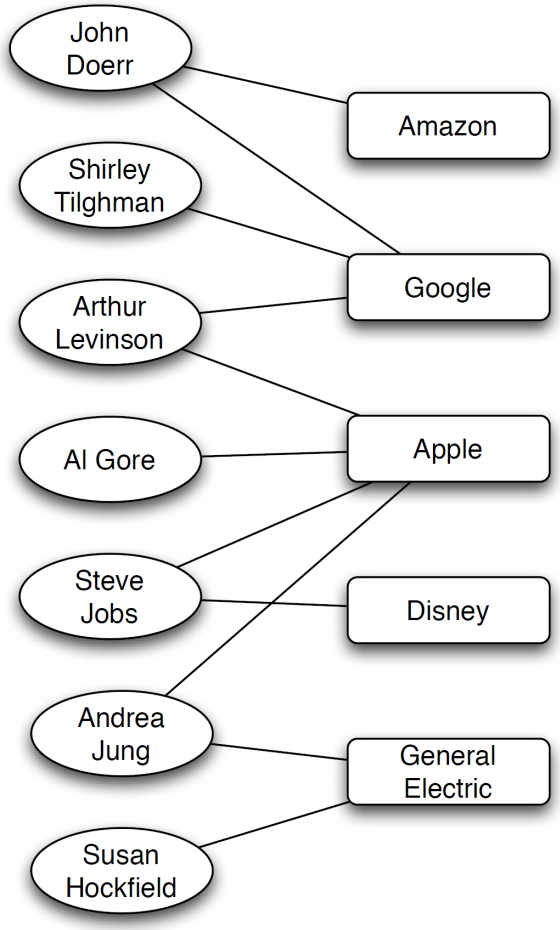
社交距离

- 讨论同质性的时候，我们介绍了归属网络
  - 二分图：左边是人，右边是社团（foci）
- 归属网络提供了同质性连边

两个人uv的社交距离 = uv同属的最小社团的大小

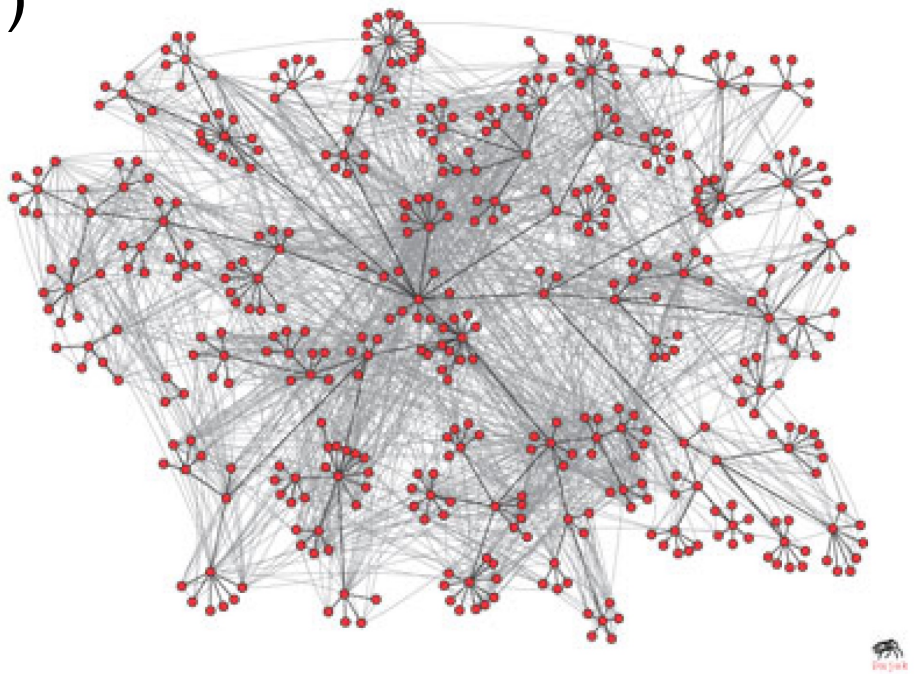
- 为何是最小社团?





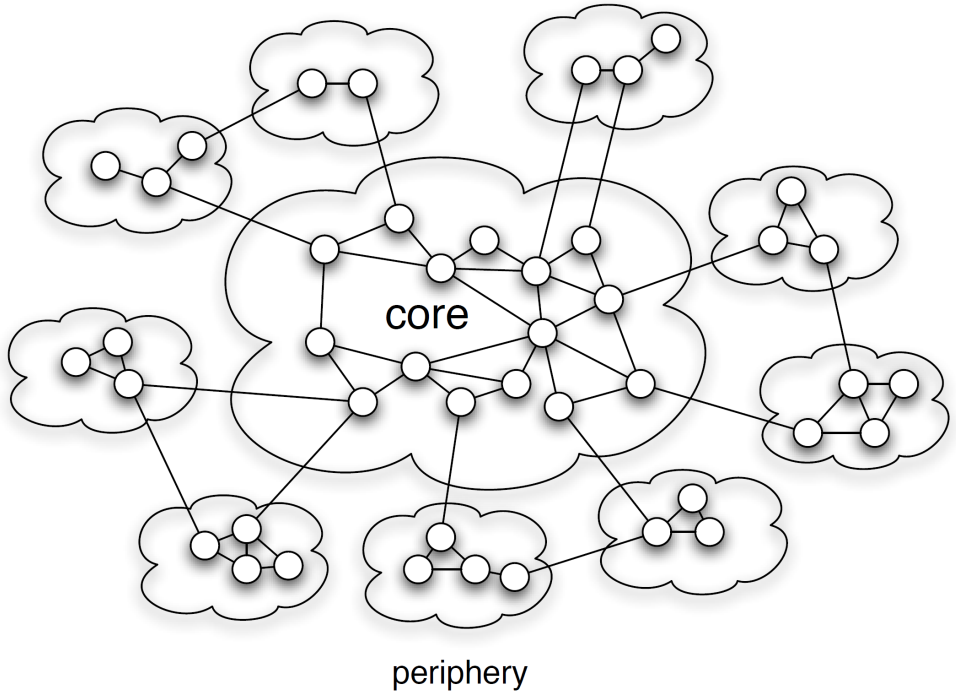
社交距离的WS模型

- 社交距离自然的定义了同质性连接
- 弱联系连边?
  - 若考虑社交距离下的WS模型，以$d(u,m)^{-q}$建立弱连边
  - q = 1是使分散搜索性能最优的 (Kleinberg, 2001)
- 实证研究:
  - 惠普员工邮件通信数据
  - 6个月内通信过一次就连边
  - 结果:
    - 若uv社交距离是d，则连边的概率约正比于~ $d^{-\frac{3}{4}}$



WS模型的问题：核心-外围结构

- 一些后续的Milgram实验表明，社会地位更高的人/名人更容易被找到
  - 解释：社会地位较高的人附近有丰富的连接，而地位低的人则很贫瘠
  - 这导致这些点的连接具有非对称性，不能用同一种概率分布来生成边
- 核心-外围结构是WS模型忽视的一种社会网络结构



计算问题：
一般图的高效距离查询

一般图的短视搜索

- 输入：无向图$G(V,E)$，|V| = n, |E| = m
- Routing scheme (1990s – 2000s较为活跃)
- 预处理：
  - 全局信息：每个节点保存一个poly log n长度的label ➡ 送信实验里面的“额外信息”
    - 住址？离哪个地标近？
  - 局部信息：每个节点保存一个尽量小的routing table
- 短视搜索：
  - 起点u终点v，利用已知信息，每步仅走邻居边并可发送poly log n长度信息
  - 性能衡量：
    - 每个点存储的信息量尽量小；需要邻居知道的信息尽量小
- 现实中最相似的应用： 网络packet的路由

相关问题： distance oracle

- Distance oracle
  - 将图预处理，建立一个数据结构
  - 之后可以高效回答任何两点间最短路距离
  - 性能衡量：预处理、回答距离所需的时间、空间尽量小
- 本质上更简单：
  - 只需要路径长度，不需要短视搜索
  - 非分布式，只关心总体上的时间、空间需求

Oracle?

计算机科学中的Oracle

- 一种理想的机器（黑盒），对于给定输入“免费”给出想要的输出
- 用于将不必要/不关心的细节隐藏：oracle的实现是另外的问题
  - Oracle access to some function，这个function可以很难evaluate或者情况复杂
- 用于抽象/规约：如果存在某oracle那么就可以XXX
  - 如ellipsoid method需要的separation oracle；类似于程序语言的“接口”
- 理想的假设：
- a random oracle is an oracle that responds to every unique query with a (truly) random response chosen uniformly from its output domain. If a query is repeated, it responds the same way every time that query is submitted

朴素方法/baseline

- Routing scheme：
  - 每个点u存完整路由表（以u开始的最短路径树/BFS树）　需要n存储
- Distance oracle：
  - n遍BFS，直接记录每对点之间的距离　需要n^2 时间

- 以上算法都是精确计算，我们聚焦于近似计算，以此换取效率

误差衡量： stretch

- 设d’是我们算法得到的距离
- 要求：d’总是更大

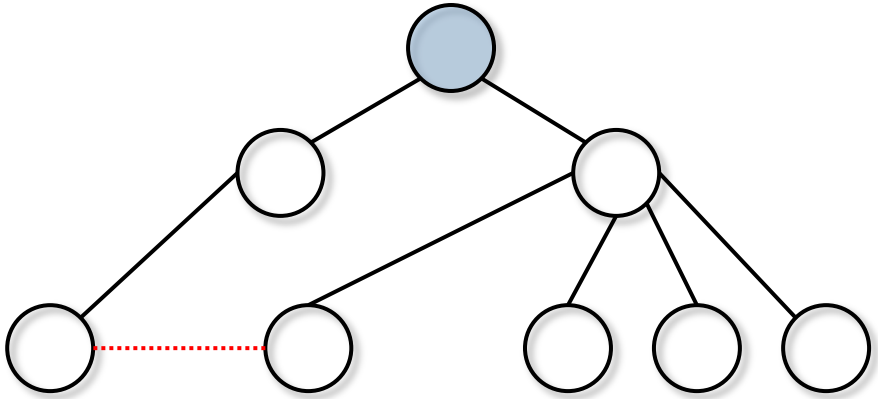
$$\text{stretch}(d') = \max_{u,v} \frac{d'(u,v)}{d(u,v)}$$

stretch至少是1，越小（接近于1）越好

stretch衡量的是每对点之间最短路的相对误差

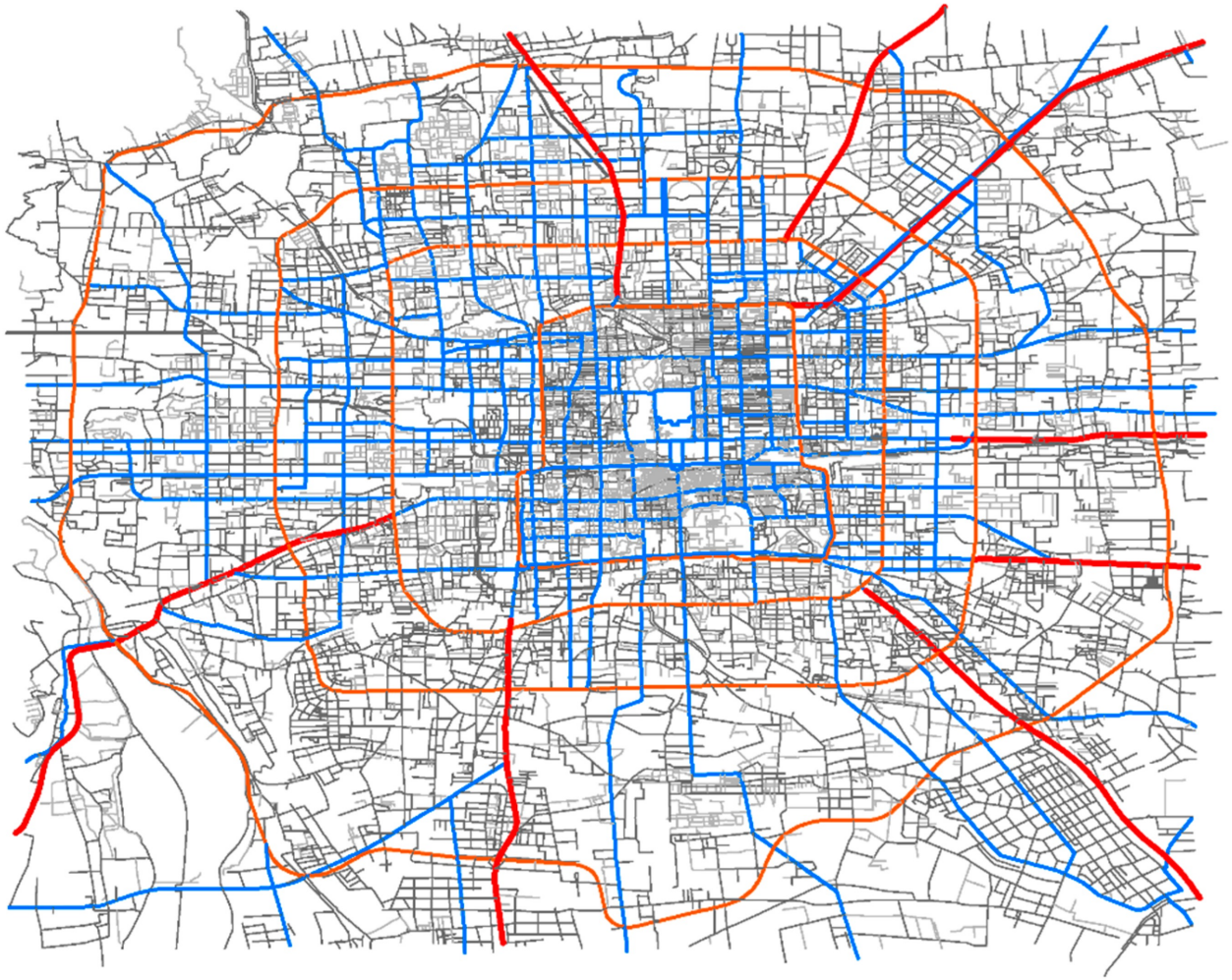
另一个极端：速度极快，误差极大

- 直接找到某个点u，然后求BFS树，然后把树上距离当作d’
  - 时间是线性的
  - 误差呢？可以到n



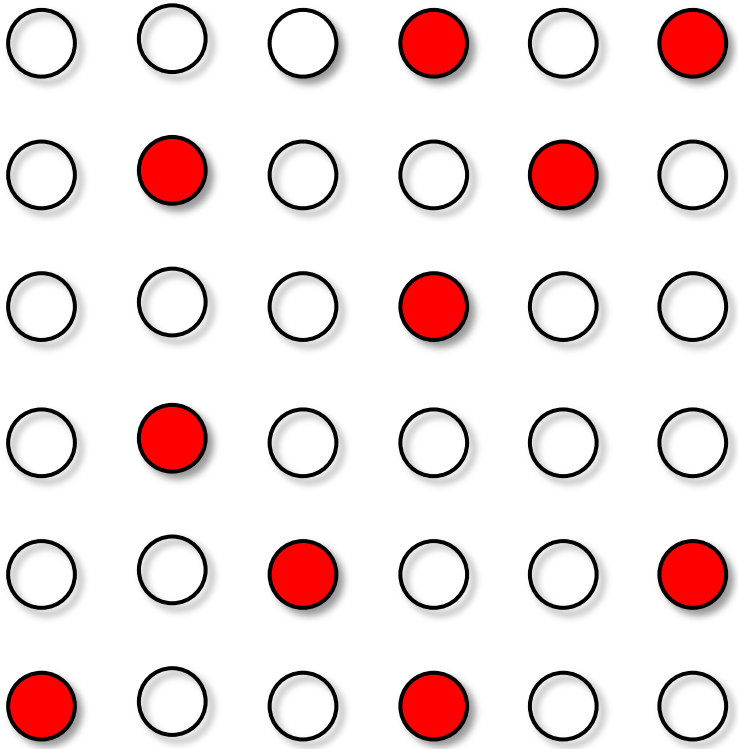
一种类比现实公路网络的思路

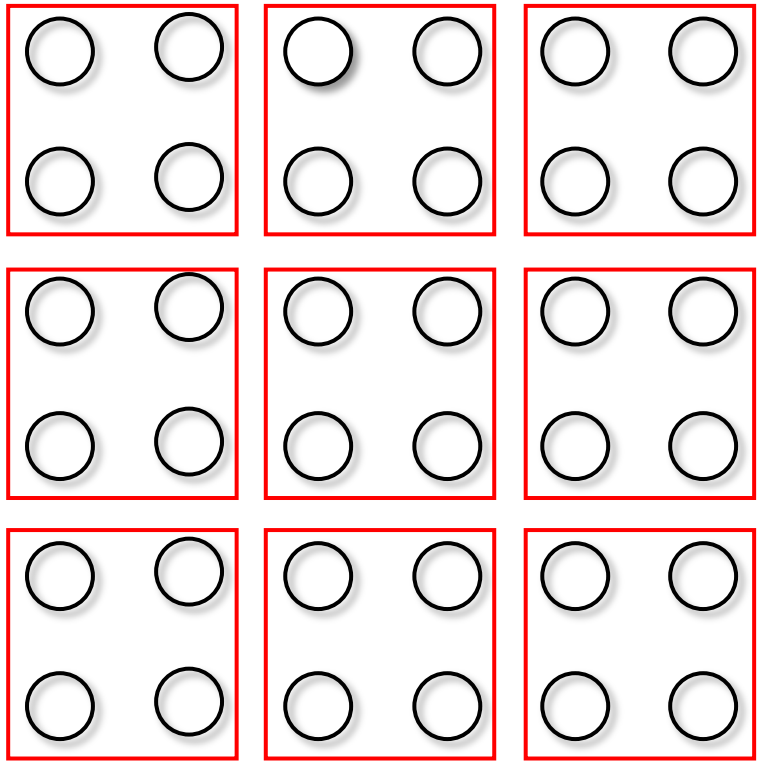
- 分层：本地连高速，高速直通互联



WS模型也是类似的

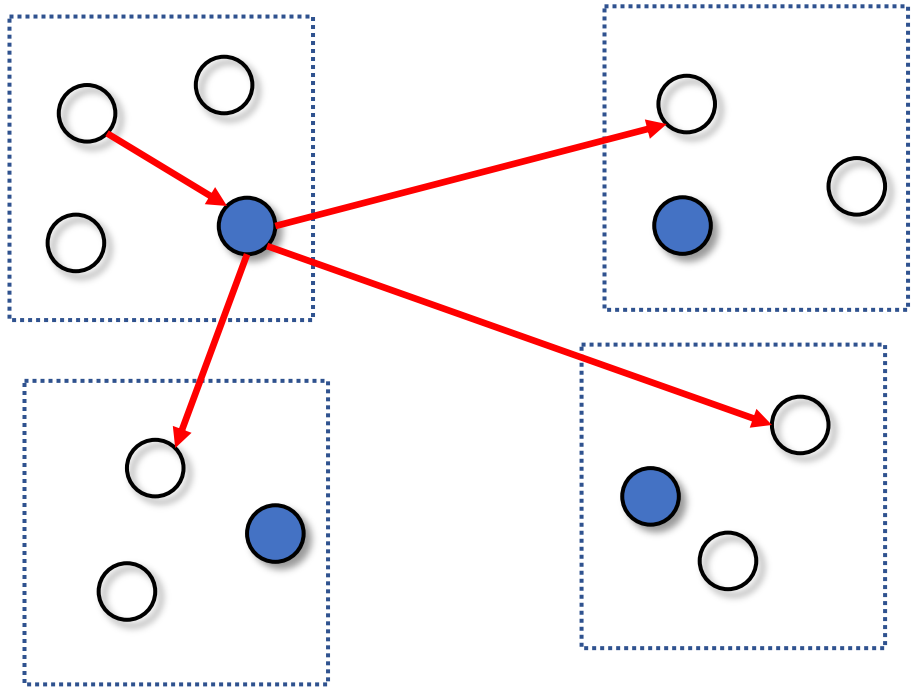
- O(log n)步之后就访问了0.5n个点S，任何未访问点到S距离是O(log n)





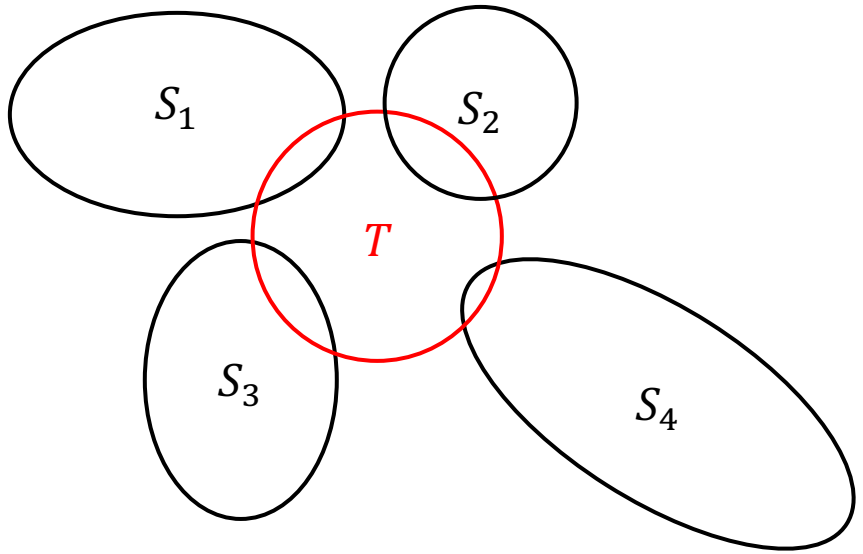
大体思路

- 考虑一个简单方法：分2层
  - 找到一些landmark
  - 所有点都有邻近的landmark
  - landmark到各个其他点都有“直接”连接
- 如果目标点v在u附近，就直接走过去
- 如果很远，就走landmark



重要工具：random hitting set

- 设有一个n元的ground set U
- 有m个集合$S_1, \ldots, S_m \subseteq U$
- Hitting set：称T集合是$S_1, \ldots, S_m$的hitting set，若$\forall i, T \cap S_i \neq \emptyset$



$S_1 = \{1,2,3,4\}, S_2 = \{1,4,5\}, S_3 = \{2,5,6\}$
$T = \{1,5\}$是一个hitting set

- 设$S_1, \ldots, S_m$每个集合都有k个元素，ground set |U| = n

  以$1 - \frac{1}{\mathrm{poly}(m)}$概率，大小为$\mathrm{L} := O\left(\frac{n \log m}{k}\right)$的随机集合T是$S_1, \ldots, S_m$的hitting set

  - T是由从U中均匀独立采样L次形成的
- 证明：固定一个S_i，

  $$\Pr[T\text{与}S_i\text{交集为空}] \leq (1 - k/n)^{\mathrm{L}} \leq 1/\mathrm{poly}(m)$$

- 相关问题对比：计算最小的hitting set
  - 计算最小的hitting set是NP-hard的！
  - 即使计算出最小，也无法说明其绝对大小L

应用

- 设U是图的节点点集
- 每个点$1 \leq i \leq n$对应的最近的k个邻居是S_i
- 那么高概率一个大小为$O\left(\frac{n \log n}{k}\right)$的随机集合T与每个$S_i$都有交
- 也就是每个点都离T很“近”

Distance oracle

- Random hitting set
  - 取k = $\sqrt{n}$，取T为大小为$O(\sqrt{n}\log n)$的随机集合；T高概率交每个点的k近邻
  - T是landmark
- Distance oracle：预处理阶段
  - 每个点u精确计算到每个S_u中的点的距离，并且同时存储S_u集合
  - 每个T中的点精确计算到其他所有n个点的距离
  - 如何有效实现？总时间、空间复杂度?
- Distance Oracle：查询阶段（给定起点u终点v）
  - u查是否$v \in S_u$，如果是，则直接返回预处理的距离
  - 如果v不在，找到T中离u最近的u'，返回

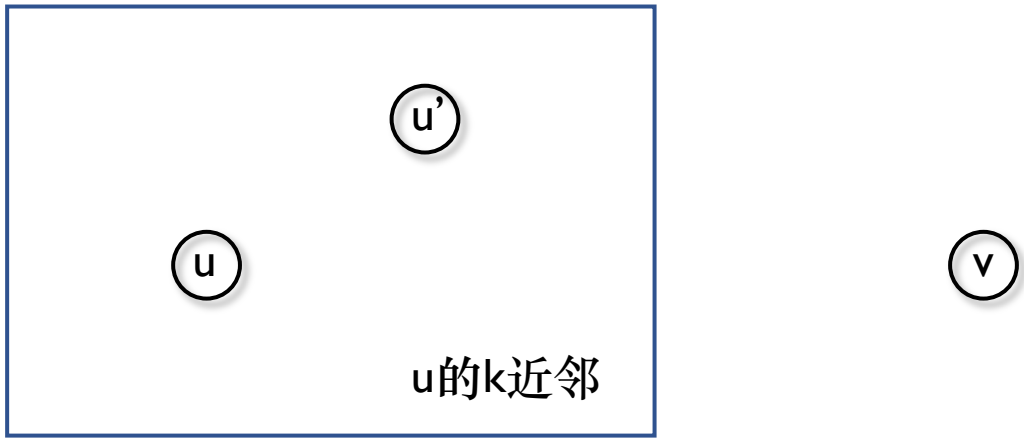
$$d(u,u') + d(u',v)$$

Stretch分析

回顾算法：

1. u查是否$v \in S_u$，如果是，则直接返回预处理的距离
2. 如果v不在，那么找到T中离u最近的u’，然后返回$d(u,u') + d(u',v)$

- 1是精确的，无误差
- 2呢？
  - v不在u的k近邻推出$d(u,u') \leq d(u,v)$
  - $d(u',v) \leq d(u,v) + d(u,u') \leq 2d(u,v)$
  - 因此$d(u,u') + d(u'v) \leq 3d(u,v)$
- stretch = 3



* 实现成Routing Scheme

如何实现成routing scheme?

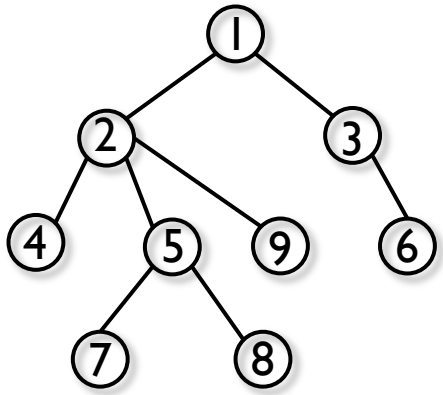
- 如何将刚才的distance oracle实现成routing scheme?
- 思路:
  - 每个点u的table保存S_u，以及u到S_u中每个点v的最短路的下一个点p(u, v)
  - 如果$v \in S_u$那么可以走到p(u, v)，最后沿着最短路走到了v
- 如果$v \notin S_u$呢?
  - 先到u的landmark $u' \in T$，然后从u'走到v
  - 每个点u额外保存到T的最近点u'，并保存p(u, u')
  - 到u'后，从u'到v怎么办?
  - 需要在u'的最短路径树上的routing scheme

（最短路径）树上的routing scheme

- 设F是一个有根树

```
DFS(u)
    label u as visited
    for every u's neighbor v that is not visited
        DFS(v)
```

- DFS深度优先搜索
  - 定义DFS序列为DFS运行中访问的点的序列
  - 性质：对于每个u，F(u)在DFS序列中是连续一段 (设F(u)代表以u为根的子树)
  - 设s(u)和t(u)是F(u)在DFS序列中的开始和结尾的位置



DFS序列： 1->2->4->5->7->8->9->3->6
F(2)对应 2->4->5->7->8->9
s(2) = 2, t(2) = 7 （2在第2位，9在第7位）

推论：给定v，可以根据s(v)是否在[s(u), t(u)]判断v是否是u的子树

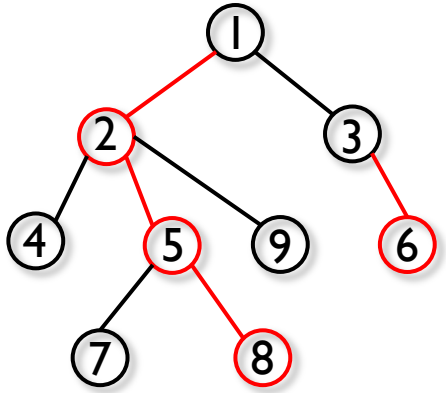
朴素的树上的routing scheme

- Label: (v, s(v), t(v))
- Table： u所有邻居v的label (v, s(v), t(v))
- Routing algorithm：
  - 如果s(v)不在[s(u), t(u)]之间，就转发给u的父亲
  - 否则找到s(v)属于哪个u的孩子z使得v在[s(z), t(z)]之间

最多要O(n)的空间！

改进的树上的routing scheme

- Heavy-light decomposition
  - 对每个节点u，称后继数量 (|F(z)|) 最多的孩子z为heavy child，其余为light child
  - 父节点连向heavy child的边为heavy edge，否则为light edge
  - 每个非叶子节点恰有一个heavy child和(连向下的) heavy edge



性质：任何节点u到树根的路径上至多有O(log n)条light edge

证明：设有light edge (u, v)那么|F(v)| <= 0.5 |F(u)|
否则|F(u)| >= |F(z)| + |F(v)| > |F(u)|

改进的树上的routing scheme

- Label: (v, s(v), t(v), v到树根路径上的所有O(log n)条light edge)
- Table of u:
  - s(u), t(u)
  - heavy child z
  - u的父节点parent(u)
- Routing algorithm (u, v):
  - 如果s(v)不在[s(u), t(u)]范围内，那么去parent(u)继续找
  - 否则
    - 如果u到某个孩子x是v到树根路径的light edge，那么走(u, x)去x继续找
    - 否则去u的heavy child z继续找

基于landmark的routing scheme

- 回忆：$\sqrt{n}$大小的随机landmark T，与所有$k = \sqrt{n}$近邻S_i相交
- 用R_u代表以u为根的最短路径树的tree routing scheme
  - $R_u(w)$表示w的table
- Label for v: (v, v最近的landmark v', $R_{v'}$上v的label)
- Table for u:
  - 存储$S_u \cup T$ ； $\forall w \in S_u \cup T$， 存储$R_w(u)$
- Routing algorithm u -> v:
  - 如果v $\in S_u$，沿v的最小路径树走（u已经存储$R_v(u)$且v是根）
    - 如果v $\in S_u$，那么沿着v的最短路径树走到的下一点x满足$v \in S_x$
  - 否则，沿v'的最短路径树走（所有点已存储$R_{v'}(u)$， $R_{v'}$上v的label是全局信息）
    - 走的时候告诉下一点也沿着v'最短路径树走，无论是否v在不在新点w的S_w里

Stretch分析

- Routing algorithm:
  1. 如果v ∈ S_u，沿v的最小路径树走（u已存储$R_v(u)$且v是根）
  2. 否则沿v'的最小路径树走（所有点已存储 $R_{v'}(u)$， $R_{v'}$ 上v的label是全局信息）
- 如果v ∈ S_u，那么沿着v的最短路径树走到的下一点x满足$v \in S_x$
  - 情况1的stretch = 1
- 另一种情况的stretch?
  - $d(u,u') \leq d(u,v)$
  - $d(v,v') \leq d(v,u') \leq d(v,u) + d(u,u') \leq 2d(u,v)$
  - $d(u,v') \leq (u,v) + d(v,v') \leq 3d(u,v)$
- stretch = 5

最优的stretch-space tradeoff

最优的stretch-space tradeoff

- 我们得到了$(2k-1)$-stretch $n^{1+1/k}$-space的distance oracle
- 下面利用Erdős关于图的girth的猜想来证明这个界时紧的
- 图的girth：图上的最短（简单）回路长度

> Erdős Girth Conjecture.
> 存在图G = (V, E)使得$|E| = \Omega(n^{1+1/k})$，并且girth(G) $\geq 2k+1$

- 验证例子：$k=1$的时候，二分完全图

利用girth conjecture证明space下界

Erdős Girth Conjecture.
存在图G = (V, E)使得$|E| = \Omega(n^{1+1/k})$，并且girth(G) $\geq 2k+1$

- 取一个边数为$\Omega(n^{1+1/k})$的girth至少是$2k+1$的图G
- Claim：G的任两个子图G_1, G_2必不能用同一个oracle达到$(2k-1)$-stretch （下页证）

- 该Claim可以推出结论
  - distance oracle: 对任意图，都能用s空间达到2k – 1的stretch
  - 那么因为有$2^{|E|}$个子图，所以必然需要$\log 2^{|E|}$个bit来区分所以$s \geq n^{1+1/k}$

Erdős Girth Conjecture.
存在图G = (V, E)使得$|E| = \Omega(n^{1+1/k})$，并且girth(G) $\geq 2k+1$

- 下证：G的任两个子图G_1, G_2必不能用同一个oracle达到$(2k-1)$-stretch
- 设有对G_1的stretch是$2k-1$的oracle A，并设oracle距离是δ
- 取$e = (u, v) \in G_1 \setminus G_2$，则$1 \leq \delta(u, v) \leq 2k-1$
- 因为girth(G)至少$2k+1$，$d_{G_2}(u, v) \geq d_{G \setminus e}(u, v) \geq 2k > \delta(u, v)$

- 即e的存在/不存在可显著改变最短路距离，而同一个distance oracle不可能同时可以处理

不用girth conjecture?

- Erdős的随机方法
  - 考虑Erdős-Renyi graph G(n, p)
  - 证明对于某个合适的p，G(n, p)中对于每个长度至多t的环删除任意一条边后，剩下边数的期望不小于$n^{1+1/(t-1)}$
  - Fact：随机图的期望大于某个边数 -> 必然存在达到期望边数的图
- 可以得到：存在图G使得girth $\geq t+1$，边数至少$n^{1+1/(t-1)}$

减弱的girth定理的证明

- 设Y_i是长度为i的环的个数，那么$\mathbb{E}[Y_i] = A_n^i \cdot \frac{p^i}{i}$
- $Y = \sum_{3 \le i \le t} Y_i$，那么$\mathbb{E}[Y] \le (np)^t$
  - $\mathbb{E}[Y_i] \le \frac{(np)^i}{i}$　　$\mathbb{E}[Y] \le \sum_{3 \le i \le t} \mathbb{E}[Y_i] \le \frac{1}{3}(np)^3 \cdot \frac{(np)^{t-3}-1}{np-1} < (np)^t$
- 设G'是G(n, p)对每个长度至多t的环删除一条边后所得的图
  - 设X是G(n, p)边数；Z = X – Y；pick p such that $\mathbb{E}[X] \ge 2\mathbb{E}[Y]$
  - $\mathbb{E}[Z] \ge \mathbb{E}[Y] = (np)^t = n^{1+1/(t-1)}$　<-　　$p = n^{\frac{2-t}{t-1}}$

相关结构：Spanner

Spanner

- Distance oracle只要求回答距离查询，不限制存储的信息类型
- 如果必须存子图呢?
- A t-spanner of a graph G is a subgraph H such that

$$\forall u, v \in V, \qquad d_H(u, v) \leq t \cdot d_G(u, v)$$

- 主要性能指标：边数

Greedy Spanner

- 推广最小生成树算法

1. sort edge in non-decreasing of weights
2. for each edge (u, v) in the sorted list
3. if $d_H(u, v) > t \cdot w(u, v)$
4. add (u, v) to H
5. return H

Stretch Analysis

- 性质：H的girth至少是t + 2
- 如果有t + 1长度回路
  - 考察最长边e，考虑e被加进去的时刻
  - 此时剩下t条边权重加起来至多t w(e)，但是这样不会加上e
- Moore's girth bound
  - A graph with girth >= 2k + 1 has at most $n^{1+1/k}$ edges
- 所以(2k-1)-stretch的greedy spanner的边数是$n^{1+1/k}$

Moore's Girth Bound

- 要证：A graph with girth >= 2k + 1 has at most $n^{1+1/k}$ edges
- 设d = 2m / n为平均度
- 迭代删除图中度 <= d / 4的节点，令得到的图为H
- H边数 >= m / 2：删除的点至多n个，每个删除d / 4边，总共删除m / 2
- H每个点的度至少d / 4
- H性质：考虑H某个点为根的BFS树，那么树必然会至少有k的深度，并且这k的深度必须是完全d / 4叉树，否则与girth是2k + 1矛盾
- 得到H边数至少是$\left(\frac{d}{4}\right)^k \leq m$，代入d = 2m / n得到m = $n^{1+1/k}$