I. ДИФФУЗНОЕ РАССТОЯНИЕ

Рассмотрим следующее семейство случайных блужданий: начинаем из стартовой вершины V_0, делаем $K \in [1, K_{\max}]$ шагов k блуждающими, в итоге получаем $kK_{\max}$ вершин размеченных некоторым значением K. Конечно же в одну и ту же вершину можно прийти разными случайными блужданиями разной длительности, однако интересно посмотреть на зависимость *диффузного расстояния* $D(V)$:

$$D(V) = \sum_{K=1}^{K_{\max}} K\tilde{p}_K(V), \qquad \tilde{\boldsymbol{p}} = \boldsymbol{p}/\|\boldsymbol{p}\|_1,$$

где $\boldsymbol{p} = (p_1(V), p_2(V), \ldots)$ – вектор из вероятностей находиться в вершине V на шаге K. Изначально нормировка такая, что $\sum_V p_K(V) = 1$.

Стоит заметить, что $D(V)$ зависит от выбора $K_{\max}$. Более того, при $K_{\max} \to \infty$ мы увидим расходимость $D(V)$ для конечных графов. Это легко увидеть, если вспомнить, что $p_\infty(V) \sim 1/N$, где N это размер графа. Тогда в сумме можем оценить $\boldsymbol{p}_\infty(V) \sim 1/K_{\max}$ и

$$\lim_{K_{\max}\to\infty} D(V) \sim K_{\max}/2 \to \infty.$$

Где наступает этот момент, когда $D(V)$ теряет информативность? Пока не могу явно оценить, но например для кубика 222 можно заметить, когда $p_K(V_0)$ выходит на константу по K (fig. 1).

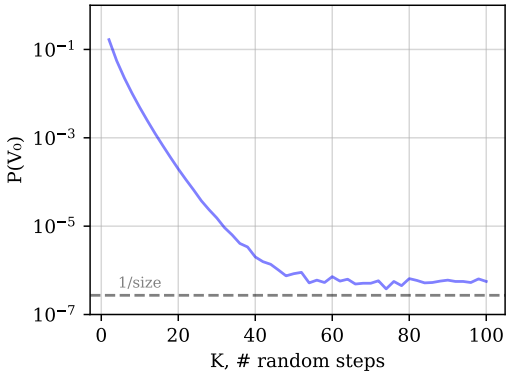

Figure 1. Вероятность оказаться в V_0 после K случайных шагов, каждая точка получена по $k = 10^8$ блуждающим

Посмотрим для меньших значений на зависимость $D(d)$, где d это истинная дистанция до V_0 (fig. 2). Для удобства переопределим $D(V_0) \stackrel{\text{def}}{=} 0$. В данном случае диаметр кубика равен 14. Видно, что для $K_{\max}$ сильно превыщающем диаметр (однако при $p_K(V) \neq \text{const}$) диффузное расстояние остаётся информативным и монотонно зависит от K (за исключением последних двух колец $d = 13, 14$, что не принципиально во время сборки кубика).

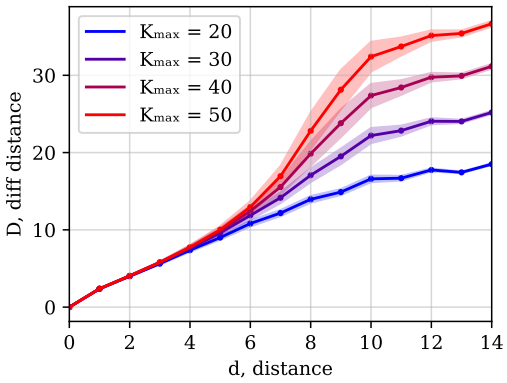

Figure 2. Зависимость диффузного расстояния D от истинного d, каждая точка получена по $k = 10^9$ блуждающим

Table I. Количество вершин в колце d для кубика 222, QTM

d	counts
0	1
1	6
2	27
3	120
4	534
5	2256
6	8969
7	33058
8	114149
9	360508
10	930588
11	1350852
12	782536
13	90280
14	276

II. CODE

Код для получения диффузного расстояния на кубике 222. Для $K_{\max} = 30$ кривая с fig. 1 считается за 2 минуты, для $K_{\max} = 50$ за 9 минут. В примере $k = 10^8$ и резудьтат усредняется $\mathrm{rep} = 10$ раз.

```python
import torch

neighbors = ... # torch.tensor
distance  = ... # torch.tensor
ds        = torch.unique(distance)
N         = neighbors.size(0)

def get_pilgrims_position(L, k=100_000):
    pilgrims_position = torch.zeros(k, dtype=torch.int64, device=device)
    for j in range(L):
        pilgrims_position = torch.gather(
            neighbors[pilgrims_position],
            1,
            torch.randint(6, (k,), device=device).view(k, 1)
        ).squeeze(1)
    return pilgrims_position

def get_D(Kmax, k=100_000_000, rep=10):
    Ks = torch.arange(1, Kmax+1, device=device)
    visited = torch.zeros((neighbors.size(0), Kmax), dtype=torch.int64, device=device)

    for R in tqdm(range(rep)):
        for (j, K) in enumerate(Ks):
            pilgrims_position = get_pilgrims_position(K, k)
            vs, counts = torch.unique(pilgrims_position, return_counts=True)
            visited[vs, j] += counts
    clear_output()

    probs = visited / torch.sum(visited, dim=1).unsqueeze(1)
    D = torch.sum(probs * Ks, dim=1)
    D[0] = 0 #D(V_0) != 0, so it is convinient just redefine it
    return D
```