

作业 1 跳表

1 理论分析

令跳表的最底层为第一层，总节点数为 n ，增长率为 p 。

在搜索时，从 $L(n)$ 层开始逐层向下搜索。如果反过来，将其看成从第一层逐层向上搜索，则搜索过程可以将搜索分为两部分。第一部分为从底层开始移动到 $L(n)$ 层。假设当前处于第 i 层节点 x 的位置，此时有两种可选路径。

- (1) 向上移动。这要求节点 x 扩展到了 $i+1$ 层，概率为 p 。
- (2) 如果节点 x 没有扩展到 $i+1$ 层，则只能向左移动，概率为 $1-p$ 。

令 $S(i)$ 为从底层至第 i 层的搜索长度，则

$$S(1) = 0$$

$$S(i) = (1-p)(1+S(i-1)) + p(1+S(i-2)) = \frac{i-1}{p}$$

则走到 $L(n)$ 层的期望搜索长度为

$$\frac{L(n)-1}{p}$$

由于 $L(n)$ 层节点的期望为 $\frac{1}{p}$ ，则接下来向左的搜索长度不会增加超过 $\frac{1}{p}$ 。
某个节点高度超过 i 的概率为 p^k ，则跳表总高度超过 k 的概率为

$$1 - (1-p^k)^n < np^k$$

取对数解得，上述概率趋于 0 时 k 趋于

$$L(n) + \frac{1}{1-p}$$

则到达 $L(n)$ 层后，向上搜索长度不会超过 $\frac{1}{1-p}$ 。

综上，可以认为搜索长度的期望上界为

$$\frac{L(n)}{p} + \frac{1}{1-p}$$

$L(n) = \log_{\frac{1}{p}} n$ ，原式可写成

$$-\frac{\log_p n}{p} + \frac{1}{1-p}$$

易知该关于 p 函数随 p 增大时先减小后增大，且 p 趋于 0 或 1 时，平均搜索长度都趋于无穷。

2 实验数据与折线图

在长度（跳表元素个数）分别为 50、100、200、500、1000，概率 p 分别为 $1/2$ 、 $1/e$ 、 $1/4$ 、 $1/8$ 的情况下，随机搜索 10000 次，得到的平均搜索长度如下。

长度 n \ 概率 p	$1/2$	$1/e$	$1/4$	$1/8$
50	10.457900	8.809000	9.055900	11.007900
100	12.722300	11.773500	12.117100	13.401500
200	12.624300	11.87810	11.323200	12.512300
500	16.483500	14.805100	14.391000	17.154300
1000	17.292900	15.695300	15.438800	20.355300

对于上表中的跳表长度和概率，理论最坏平均搜索长度如下。

长度 n \ 概率 p	$1/2$	$1/e$	$1/4$	$1/8$
50	13.287712	12.215958	12.621046	16.193140
100	15.287712	14.100127	14.621046	18.859807
200	17.287712	15.984297	16.621046	21.526474
500	19.931569	18.475033	19.264902	25.051615
1000	21.931569	20.359202	21.264902	27.718282

在跳表长度不变情况下，相应实际情况、理论情况下平均搜索长度和增长率 p 的关系折线图如下。

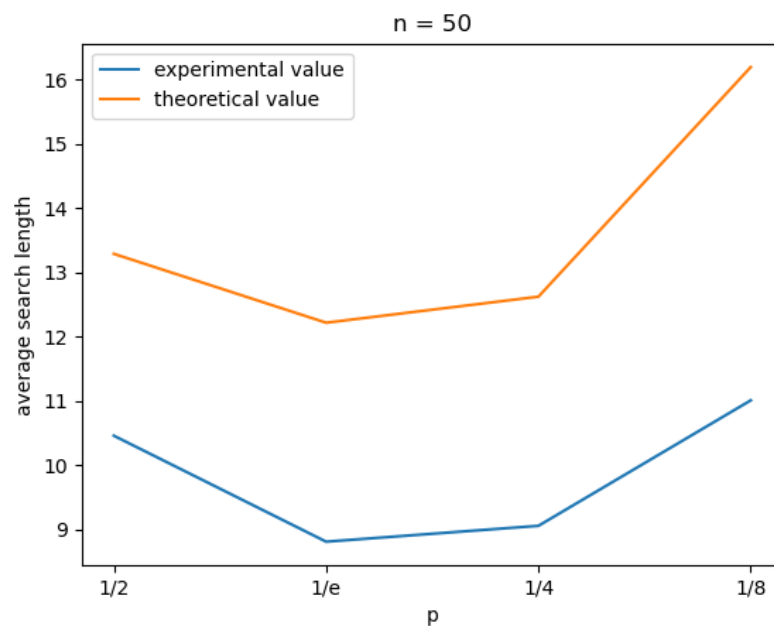


图 1: $n = 50$

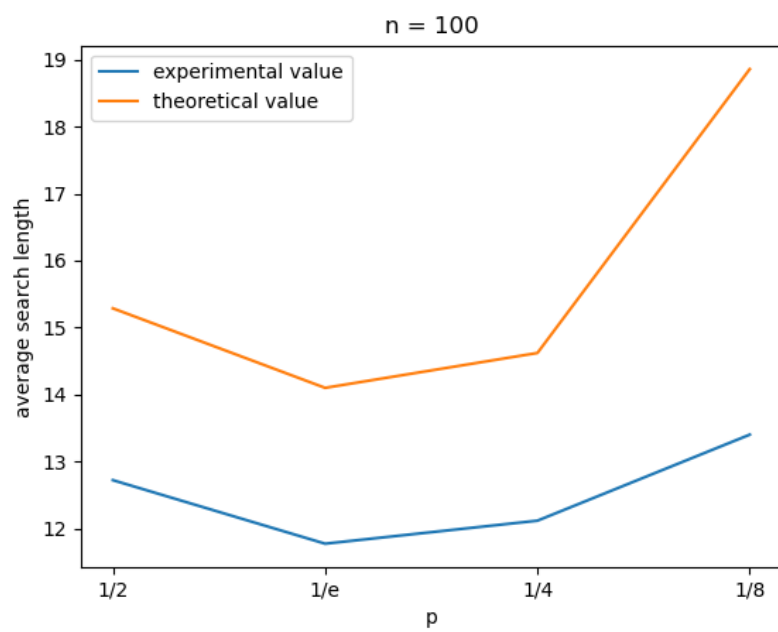


图 2: $n = 100$

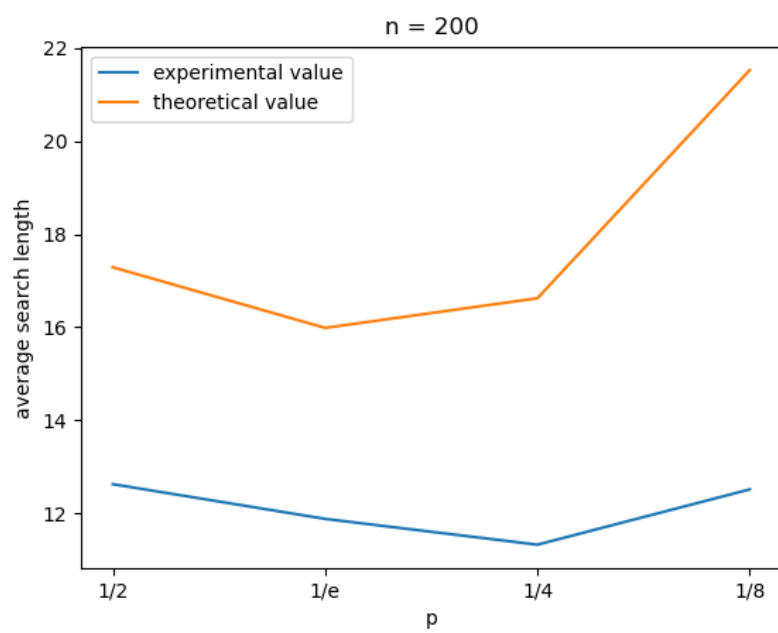


图 3: $n = 200$

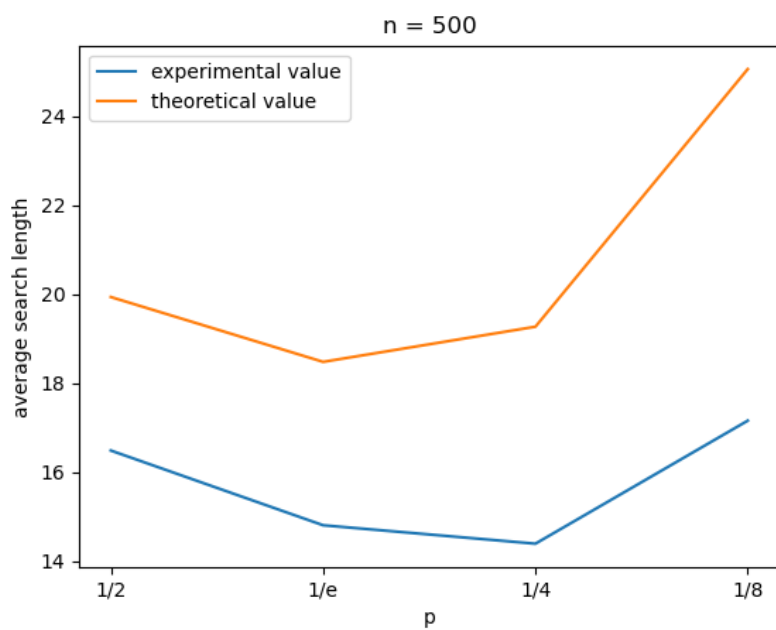


图 4: $n = 500$

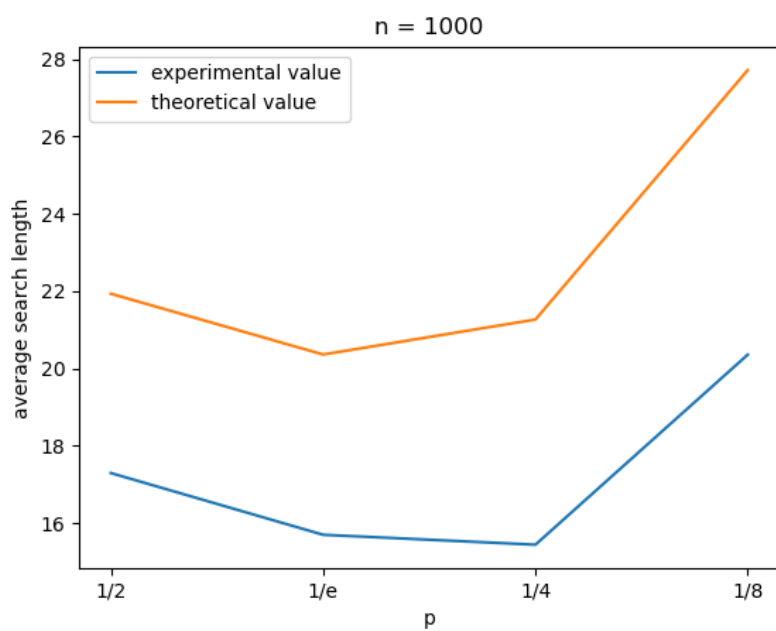


图 5: $n = 1000$

3 实验数据分析

图中可以看到，无论 n 取何值，折线图中平均搜索长度都随增长率 p 的增大而先减小后增大，且理论曲线和实际曲线走势相同，证明了理论分析中结论正确。此外，基本上平均搜索长度随 n 的增大而增大。

此外，实验计算的是随机种子下的平均搜索长度，而理论公式给出的是最坏情况下平均搜索长度的期望，所以理论平均搜索长度均大于实验平均搜索长度。此外，由理论曲线得 p 为 $1/e$ 时最优。

4 绘图代码

```
import matplotlib.pyplot as plt
import numpy as np

ps = ["1/2", "1/e", "1/4", "1/8"]
ns = [50, 100, 200, 500, 1000]

experiment = [[10.457900, 8.809000, 9.055900, 11.007900],
               [12.722300, 11.773500, 12.117100, 13.401500],
               [12.624300, 11.87810, 11.323200, 12.512300],
               [16.483500, 14.805100, 14.391000, 17.154300],
               [17.292900, 15.695300, 15.438800, 20.355300]]

theory = [[13.287712, 12.215958, 12.621046, 16.193140],
           [15.287712, 14.100127, 14.621046, 18.859807],
           [17.287712, 15.984297, 16.621046, 21.526474],
           [19.931569, 18.475033, 19.264902, 25.051615],
           [21.931569, 20.359202, 21.264902, 27.718282]]

for i in range(5):
    n = ns[i]
    plt.plot(ps, experiment[i], label = "experimental value")
    plt.plot(ps, theory[i], label = "theoretical value")
    plt.xlabel("p")
    plt.ylabel("average search length")
    plt.legend()
    plt.title("n = {}".format(n))
    plt.show()
    plt.savefig("./graph/{}.png".format(n))
    plt.close()
```