

数据结构实验报告

姓名： 杨家玺 学号： U201717007 班级： 软工 1703 班

实验一 求整数和、切 pizza 和 Hanoi 塔等问题的求解

一、实验描述

用 C 语言编程实现求整数和，切 pizza 以及 Hanoi 塔等问题的求解，在程序中加入 `clock()` 来计算求解时间，使用不同的输入值得到对应的时间值。分析算法的时间复杂度并与测量结果进行比较，如果存在差异，解释原因。

二、实验之前

1. 程序计时方法：在算法执行前于执行结束后分别执行 `clock()`，获取算法的运行 ticks，不除以 `CLOCK_PER_SECOND`，因为个人认为使用 ticks 会更加准确。
2. 设计了三个宏指令以简化程序
3. 计时原则：同一规模多次试验取平均，规模渐进增长
4. 算法复杂度绘图：绘制规模-ticks 图，绘制 $f(\text{规模})$ -ticks 图， f 为关系函数。
5. 使用 Python 程序辅助统计并进行画图，部分代码如下：

```
def run_test(cmd, size, method):
    res = {}
    for mth in method:
        res[mth] = {"size": [], "ticks": []}
        for sz in tqdm(size):
            cmd_s = cmd + " " + str(sz) + " " + str(mth)
            tmp = 0
            times = 5
            for _ in range(times):
                tmp += int(subprocess.check_output(cmd_s.split(" ")).decode("utf-8"))
            res[mth]["size"].append(sz)
            res[mth]["ticks"].append(tmp / times)  # 5次取平均
        res[mth]["size"] = np.array(res[mth]["size"])
        res[mth]["ticks"] = np.array(res[mth]["ticks"])
    return res

# 参数 cmd2 为命令行中执行“可执行文件”的命令，例如 ./a.out
res = run_test(cmd2, np.arange(500, 100000, 100), [0])
show(res[0]["size"], res[0]["ticks"], "Integer Sum Problem", lambda t: t)
# t -> f(t) 为 size 的变换关系，这个变换使 f(size) 与 ticks 呈现线性关系
# e.g.      三次方      : lambda t : t**3
           对数        : lambda t : np.log(t)
           nlogn       : lambda t : t * np.log(t)
```

三、实验设计

1. 整数求和问题：计算小于等于 n 的整数的和， n 取 500~100000 每隔 100 取一个。
2. 切 pizza 问题：利用递归算法计算一块 pizza 在切 n 刀后最多可产生多少块小 pizza。
3. 河内塔问题：利用递归算法计算规模为 n 的汉诺塔需要的步数。
4. 使用上一步定义的三个宏获取核心算法执行所需的 ticks
5. 利用 Python 自动完成程序的调用与绘图输出

四、实验过程

1. 整数求和问题

- (1) 在 `argv` 中获取 `Python` 传递来的规模参数，赋值给 `n`
- (2) 调用 `solve(n)` 通过递归求解整数和问题
- (3) 同一规模执行 5 次，取平均
- (4) 在 10 个 10 分位点抽取数据填入表格
- (5) 使用所有数据绘制散点
- (6) 使用最小二乘法拟合直线

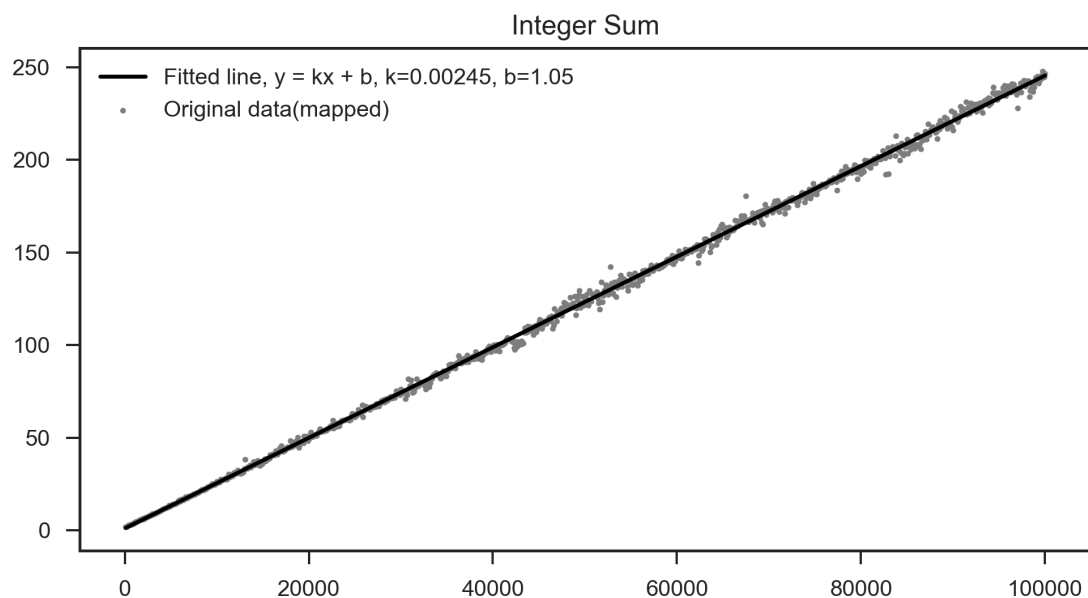
2. 切 pizza 问题，同上

3. 河内塔问题，同上

五、实验结果

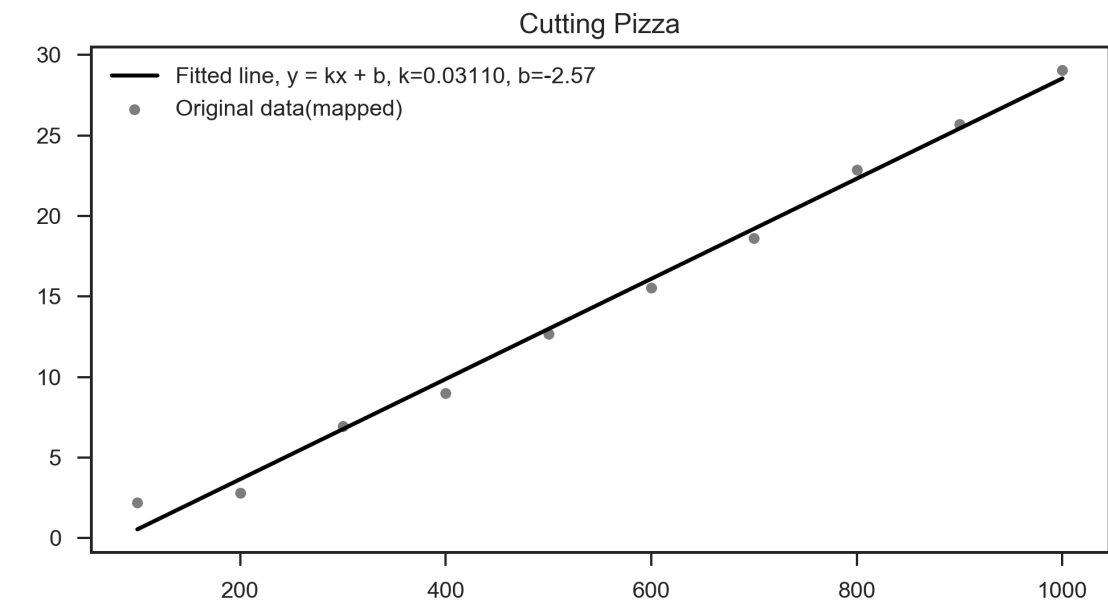
1. 整数求和问题

n	100	10100	20100	30100	40100	50100	60100	70100	80100	90100
f(n)	100	10100	20100	30100	40100	50100	60100	70100	80100	90100
ticks	1.9	25.85	49.9	75.15	99.95	124.7	147.25	172.95	193.	225.6



2. 切 pizza 问题

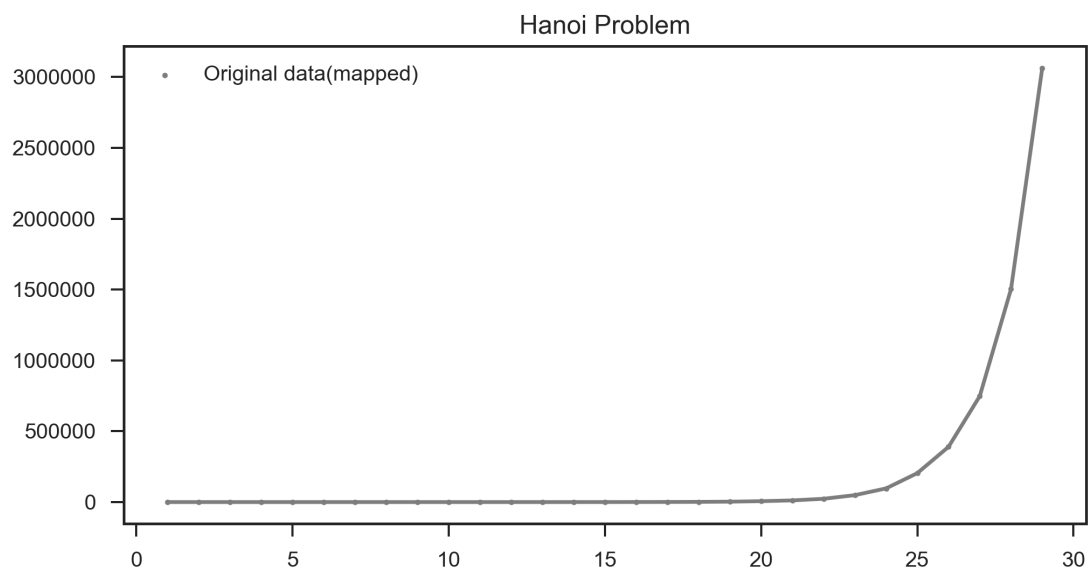
n	100	200	300	400	500	600	700	800	900	1000
f(n)	100	200	300	400	500	600	700	800	900	1000
ticks	2.208	2.816	6.954	8.982	12.666	15.528	18.614	22.86	25.704	29.066



3. 河内塔问题

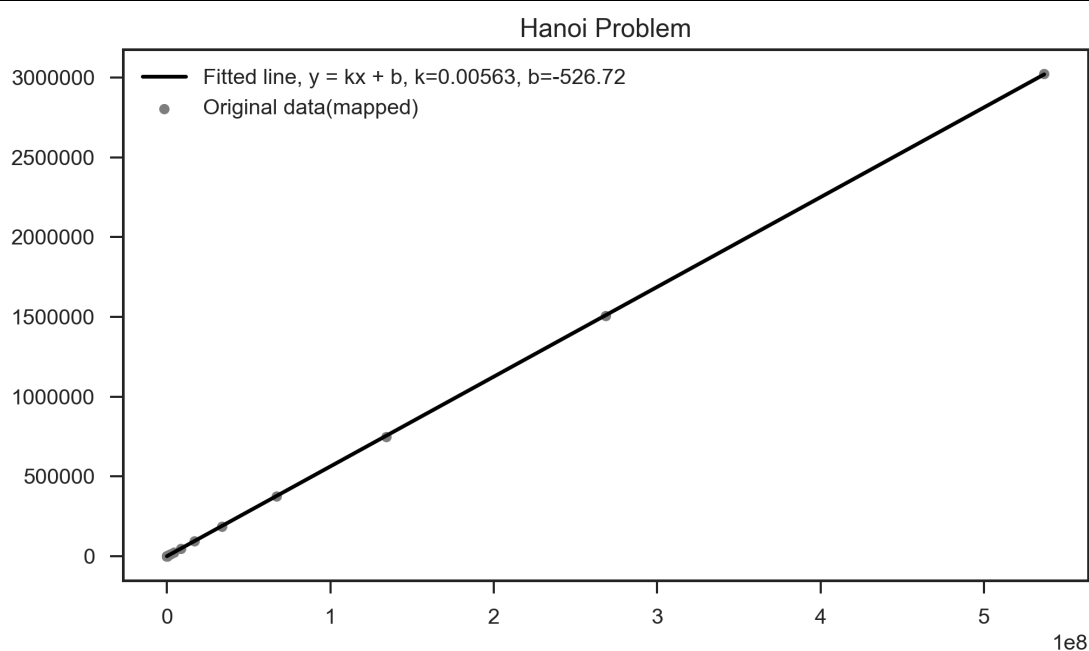
(1) 直接绘制

n	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
f(n)	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
ticks	2.0	2.0	2.2	2.8	4.6	13.4	50.2	205.4	765.2	2918.6	11745.2	48818.8	204185.2	747838.8	3064255.0



(2) 曲线拟合

n	$f(n)=2^n$	ticks	n	$f(n)=2^n$	ticks
1	2	1.80	17	131072	725.00
3	8	1.60	19	524288	2902.40
5	32	2.20	21	2097152	11636.60
7	128	2.20	23	8388608	46599.20
9	512	4.60	25	33554432	186975.20
11	2048	14.40	27	134217728	747769.80
13	8192	49.20	29	536870912	3025041.60
15	32768	183.00			



六、实验结论

1. 问题时间复杂度分析

(1) 求整数和与切 pizza 问题均为线性时间复杂度问题，时间复杂度 $T(N)=O(N)$

(2) 汉诺塔问题使用递归算法，递归步有 $T(N)=2*T(N-2)+1$ ，平凡情况为 $T(1)=1$ ，求解得到 $T(N)=2^N-1$ ，故 $T(N)=O(2^N)$

2. 理论分析与测量结果的对比

(1) 求整数和问题与切 pizza 问题的最后得到的真实结果的散点图均匀分布在拟合直线的两侧且波动很少，考虑机器每次运算产生的差异，可认为实验结果基本吻合理论分析。

(2) 河内塔问题的结果曲线基本吻合理论分析，观察 2^n 与运行 ticks 的线性曲线可知理论分析的正确性。