数据结构与算法作业二

Graphs, recursion, and invariants

课程：数据结构　姓名：Your Name　学号：00000000

要求：写出关键步骤和复杂度；可以使用英文或中文解释，但请保持符号定义一致。

Problem 1　递归与复杂度

设 $T(n) = 2T(n/2) + n$，其中 $T(1) = 1$。使用递归树说明 $T(n)$ 的渐近复杂度，并解释每一层的工作量。

Solution　第 i 层有 2^i 个子问题，每个子问题的非递归工作为 $n/2^i$，因此这一层的总工作量为 n。树高为 $\log_2 n$，叶节点总工作量为 n，所以

$$T(n) = \Theta(n \log n).$$

这也与主定理中 $a = 2, b = 2, f(n) = n$ 的结论一致。

Problem 2　最短路径

给出一个使用优先队列的 Dijkstra 算法框架，并说明非负边权是算法正确性的必要条件。

Solution　将源点的距离设为零，其余距离设为无穷大。每次从优先队列取出当前距离最小的未确定顶点 u，再对每条边 (u, v) 执行松弛：若 $d(v) > d(u) + w(u, v)$，则更新 $d(v)$ 并重新加入队列。因为边权非负，已经取出的最小标签不会被未来路径改善；若存在负边，这个单调性可能失效。

Problem 3　证明不变量

为下面的循环写出一个不变量，并用三句话证明它保持成立：

```
for i = 1,…,n:      s ← s + a_i.
```

定理 1. *若循环开始前 $s = 0$，循环结束后 $s = \sum_{i=1}^{n} a_i$。*