

北京交通大学软件学院
M210004B - 算法设计与分析 (2024 年春)
书面作业 4

这是本课程的第四次书面作业，占总成绩的 2%，对于下面的证明题你需要写出详细的证明过程（即不可以出现诸如“显然”、“易得”等论证方式），对于所有的计算题需写明计算步骤，**只列出答案的不得分**。在提交作业时，你**只能提交 PDF 版本**，推荐使用 L^AT_EX 对作业进行排版（课程主页上有一些教程供参考，你也可以自己查看其它的相关教程），你也可以使用 MS-Word 或者 WPS 排版，但请不要提交排版的源文件（tex 文件或者 doc/docx 文件）。**未按要求提交 PDF 格式的作业将不被评阅，直接记为 0 分！**此外，你被允许与其他同学讨论课程的作业内容，但是一旦当你开始书写作业时，所有你将提交的作业内容都必须由自己独立完成。请参见《教学大纲实施细则》了解有关学业诚信的特别规定。如果你不清楚你的行为是否违反了有关学业诚信的特别规定，请**先**通过邮件、讨论区的方式与课程组取得联系。本次作业的截止时间为 2024 年 4 月 30 日 00:00:00，也就是说你需要在 2024 年 4 月 29 日 23:59:59 前将作业提交至 KL 教学平台。

1. (10 分) **计算建堆的时间复杂度**：已知堆是一种基于完全二叉树的数据结构（如果你忘记了，请复习一下数据结构），除了最底层外，其余各层都是满的，且均是从左向右填充。一个最大堆（又叫作大顶堆）的特点为对于每个结点而言，其子节点均小于该节点。本题使用数组表示一个堆，该数组用 $A[1..n]$ 来表示，即有 n 个元素，元素下标由 1 到 n 。该数组的根节点为数组下标为 1 的元素，对于每个结点，若其下标为 i ，则其左儿子结点 $LEFT(i) = 2 \times i$ ，其右儿子结点 $RIGHT(i) = 2 \times i + 1$ 。图 1 是一个最大堆的例子，左半部分表示其逻辑结构，右半部分表示其物理存储。

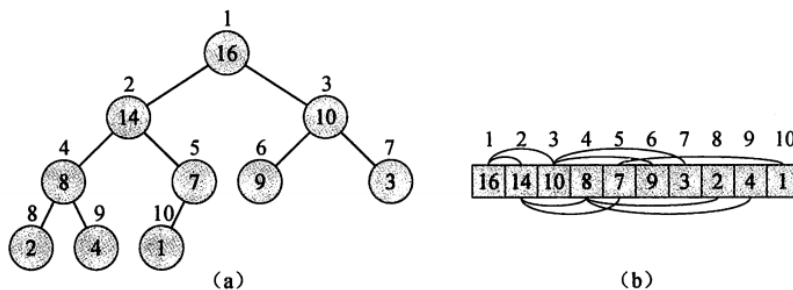


图 1: 一个最大堆

本题要求你考虑将一个数组 A 原地调整为一个最大堆，为了实现该功能，你需要使用如下 MAX-HEAPIFY 函数，该函数在假设由数组第 i 个元素表示的结点它们的子结点作为根节

点的子树分别都是最大子堆的前提下，将由数组第 i 个元素表示的结点作为根结点的子树调整为一个最大子堆：

MAX-HEAPIFY(A, i)

```

1:  $l = \text{LEFT}(i)$ 
2:  $r = \text{RIGHT}(i)$ 
3: if  $l \leq n$  and  $A[l] > A[i]$  then
4:    $largest = l$ 
5: else
6:    $largest = i$ 
7: if  $r \leq n$  and  $A[r] > A[largest]$  then
8:    $largest = r$ 
9: if  $largest \neq i$  then
10:   exchange  $A[i]$  with  $A[largest]$ 
11:   MAX-HEAPIFY( $A, largest$ )

```

图 2演示了其调用过程。

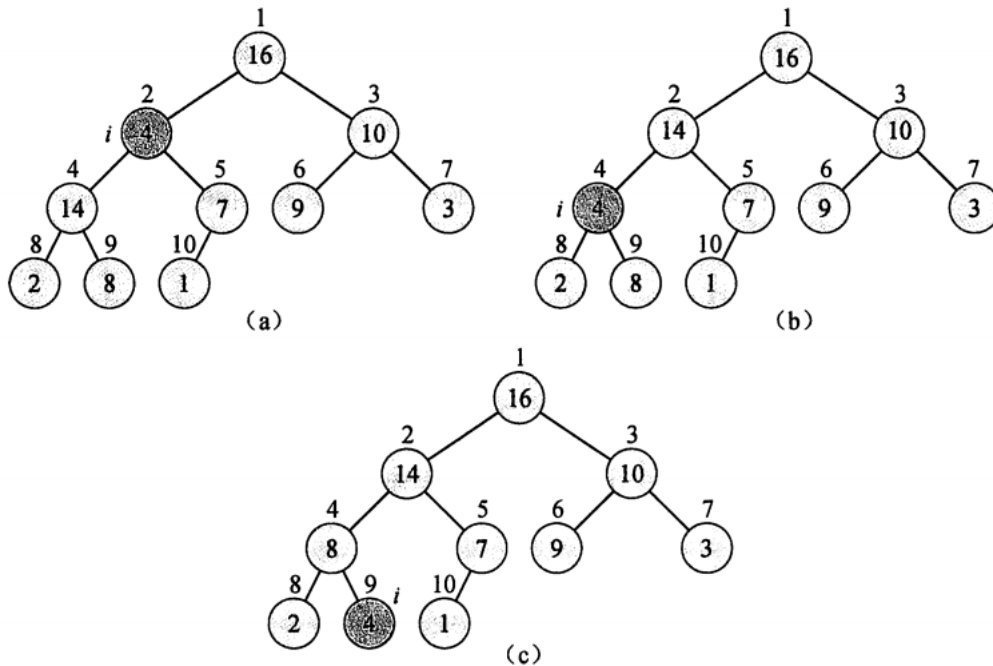


图 2: 调用 MAX-HEAPIFY($A, 2$) 的过程示例：在调用前，即图 (a)，由 $A[2]$ 表示的结点其左子树和右子树已经分别为一个最大堆，此时结点 $A[2] = 4$ 违反了最大堆原则，因此要进行调整，将其下沉为图 (b)，进而下沉为图 (c) 表示的状态。

此时，若要将整个数组 A 调整为最大堆，则可执行 BUILD-MAX-HEAP 函数来完成：

BUILD-MAX-HEAP(A)

```
1: for  $i = \lfloor n/2 \rfloor$  downto 1 do  
2:   MAX-HEAPIFY( $A, i$ )
```

该函数从倒数第二层开始自右向左逐个将每个结点及其子树调整为最大堆。

请证明，对于含有 n 个元素的无序数组 $A[1..n]$ ，调用 BUILD-MAX-HEAP(A) 将其调整为最大堆的时间复杂度为 $O(n)$ 。

(本题对于各位同学而言可能有些困难，因此我允许大家寻求任何帮助解决该题目，不扣分且不扣除参考额度。)

2. (10 分) 有正实数构成的数字三角形如下图所示，第一行的数为 a_{11} ，第二行的数从左到右依次为 a_{21}, a_{22}, \dots ，第 n 行的数为 $a_{n1}, a_{n2}, \dots, a_{nn}$ 。从 a_{11} 开始，每一行的数 a_{ij} 只有两条边可以分别通向下一行的两个数 $a_{(i+1)j}$ 和 $a_{(i+1)(j+1)}$ 。请设计一个算法，计算出从 a_{11} 通到 $a_{n1}, a_{n2}, \dots, a_{nn}$ 中某个数的一条路径，并且使得该路径上的数之和达到最大。

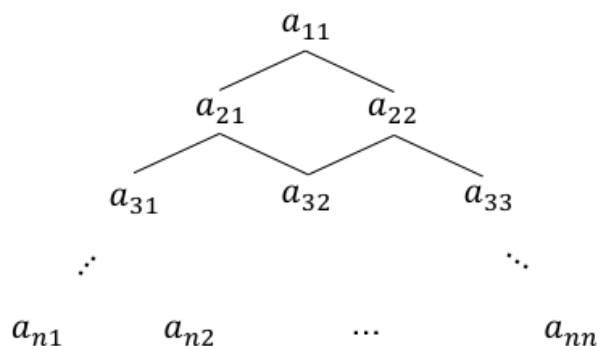


图 3: 数字三角形

3. (20 分) **小偷问题**：有一个小偷来到一条商业街，这条商业街自东向西有 n 个商家，一家挨着另一家。一天晚上，小偷在开始了他的偷窃计划。对于这 n 个商家，每个商家都存有一定数量的钱财，记第 i 个商家存有的钱财为 a_i 。小偷一旦确定偷窃这个商家，就把它的所有钱财全部偷走，但是，如果小偷连续偷了两个相邻的商家，即第 i 个商家和第 $i+1$ 个商家，例如偷了第 2 个商家和第 3 个商家，警报就会自动响起，此时，我英勇的人民公安队伍就会挺身而出将其捉拿归案！现在请问，小偷在不被警察叔叔抓走的前提下，最多能偷多少钱？

请写出该问题中存在的最优子结构，并根据该最优子结构设计一个动态规划算法，说明你所设计算法的正确性。