清华大学计算机系

2022 年“大中衔接”研讨与教学活动

第二试

时间：2022 年 5 月 22 日 09:00 ~ 13:00

题目名称	任务 1：网页解析	任务 2：网络爬虫	任务 3：BM25 计算	任务 4：PageRank 计算	任务 5：基于决策树的文本分类
题目类型	传统型	传统型	传统型	传统型	传统型
输入	标准输入	标准输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	4	4	4	4	4
测试点是否等分	是	是	是	是	是

任务 1：网页解析（16 分）

【题目背景】

HTML 技术在互联网环境中获得了广泛的使用，我们平常浏览的各种网页一般都是基于 HTML 解析生成的。因此，HTML 网页解析，作为后续网页文本分析的重要一环，在信息检索领域发挥着重要作用。请阅读《学习手册》中 HTML 相关知识，对于给定的 HTML 网页内容，将其中包含文本内容，按照 HTML 文档中出现的先后顺序，打印输出各项文本内容以及他们所处的文档结构中的位置。

为了确保测试样例不过于复杂，我们对数据做了如下这些约束和解析约定：

1. 数据保证每个测试样例的文本长度不超过 100,000 字符；
2. HTML 标签只包含 `<html>`, `<div>`, `<p>`, `<span>`, `<img>`, `<style>`, `<script>`, `<a>`, `<ol>`, `<li>`, `<h2>`。其中`<style>`、`<script>` 和 `<img>` 标签不包含有效文本内容，在解析时可直接忽略；
3. HTML 标签内可能会包含若干属性值，例如 `<a href="http://baidu.com" hidden target="_blank">baidu</a>`，例子中的 `href`, `hidden` 和 `target` 均为标签 `<a>` 的属性值，这些属性值在解析过程中不会对输出文本内容产生影响，数据保证属性值与属性值之间、属性值与标签名之间是由唯一的一个空格隔开，在 < 与标签名之间、> 与标签名（或属性值）之间、标签名和属性值内部均不包含空格等不可见字符。此外，标签的终止符内则不包含不可见字符；
4. 打印输出时，我们仅输出包含有效文本内容的标签及其所包含的内容。不包含有效文本内容的情形包括两类情形：
    1. 标签类别为 (2) 中所列的 `<style>`, `<script>` 和 `<img>` 其中之一；
    2. 标签内所包含的文本内容为空，或者仅由空格符、制表符（'\t'）和换行符（'\n' 和'\r'）构成。
5. HTML 标签内的内容可能会被子标签隔，在解析过程中，考虑到我们需要按照 HTML 文档中出现的顺序打印各项文本内容，因此我们需要把子标签隔断前后的文本内容单独作为两个部分打印输出，两者的标签内容保持一致。例如 `<div>foo<script>bar</script>dag<span>cat</span>baz</div>`，输出时需要将 `foo`, `cat` 和 `baz` 作为三条结果输出，其中还有子标签 `<span>` 包含有效文本内容，因此输出顺序为 `div:foo`, `div:dag`, `div>span:cat`, `div:baz`。
6. HTML 标签内的文本内容中，不包含 < 和 > 符号，因此你可以放心地将 < 和 > 符号作为标签名的边界来处理；
7. 所给的 HTML 文档包含若干行,数据保证它们均以 `<html>` 作为开始,以 `</html>` 作为终止，数据同时保证除了 `<img>` 标签外，所有标签均包含与之对应的终止标签（例如标签 `<a>`，在改标签内容结束后会包含与之对应的终止符 `</a>`）。

【输入格式】

从标准输入读入数据。

输入包括若干行，均为单个 HTML 文档的源码。

【输出格式】

输出到标准输出。

输出若干行，每一行输出一段解析的文本内容在网页中所处的结构位置以及具体的文本内容信息，两部分之间用英文冒号（:）隔开，网页结构位置按照从外到内的顺序展示，标签之间用大于号（>）分隔。例如在 `<html>` 标签的 `<div>` 子标签下面的一个 `<p>` 标签内包含文本内容 `hello, world`，则输出 `html>div>p:hello, world`。

注意，对于非文本内容信息，或文本内容为空的部分，不需要输出。

【样例 1 输入】

```
<html>
<div id="main">
    <h2>This is a title.</h2>
    <span onclick="check();">  </span>
    <img src="/images/test.jpg">
    <p> <a href="#">Look!</a>This is a paragraph.</p>
</div>
<script type="javascript">
function check(){
    alert("Error occurs.")
}
</script></html>
```

【样例 1 输出】

```
html>div>h2:This is a title.
html>div>p>a:Look!
html>div>p:This is a paragraph.
```

【样例 1 解释】

在上述样例中，`<script>` 和 `<img>` 标签为非文本内容信息，`<html>`、`<div>`、`<span>` 标签以及 `<p>` 标签被 `<a>` 标签分割的前半段文本内容中仅包含空格、换行符和制表符，

因此都不需要输出。只需要输出 <h2>、<a> 标签以及 <p> 标签被 <a> 标签分割的后半段的文本内容。

【子任务】

共包含 4 个测试样例，其中仅包含 ASCII 码字符，其中不可见字符仅包含空格符、制表符（'\t'）和换行符（'\n' 和'\r'）。不含中文等内容。

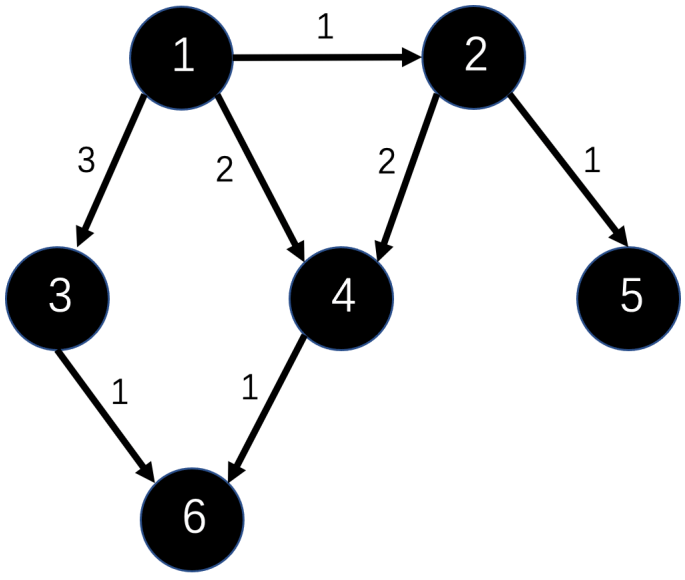
任务 2：网络爬虫（17 分）

【题目背景】

网络爬虫已经运用到了互联网应用的方方面面。请阅读《学习手册》中有关网络爬虫、深度优先搜索相关材料。在给定网页文档集合、初始种子 URL 集合的情况下，我们希望你能实现一个网络爬虫调度程序，模拟爬虫抓取解析网页的过程。

你需要实现的网络爬虫采用的是深度优先策略进行网页抓取，爬虫将会**根据初始种子 URL 集合的大小，开设同等数量的网页爬取解析线程**同时爬取网页数据。假设每个线程爬取和解析网页所花费的时间是一致的。除此之外，爬虫系统还会开设一个任务调度线程管理这若干个网页爬取解析线程的工作。在每一轮任务调度过程当中，任务调度线程会按照初始种子 URL 集合中所给的顺序依次分配给各网页爬取解析线程本轮需要爬取解析的网页编号。当然，为了确保资源的利用效率，任务调度线程会在分配前确认待爬取解析网页是否已被分配，如果已被爬取则跳过该网页至下一个未被爬取的网页。由于我们的任务调度线程还不够智能，因此它仅会去核查各线程待爬取的网页是否已被分配，而不会将其他线程未完成的任务分配给空闲的线程。当某个线程已经完成了所有的抓取任务，则该线程会结束，等待其他线程完成任务。

为了更清晰地说明网络爬虫调度过程，这个地方我们以 $\{1,2,3,4,5,6\}$ 作为网页文档集合，$\{1,2\}$ 作为初始种子 URL 集合为例进行举例说明。下图展示的是示例网页文档之间的链接关系：



图中黑色的圆圈表示的是网页的编号，箭头表示网页之间的链接关系，箭头上的数字表示链接在文档中出现的顺序。例如图中结点 1 三条指出的箭头表示网页 1 中引用了 3 个网页的链接，按照先后出现顺序分别为网页 2、4、3。下面我们展示这个例子当中爬虫的调度过程：

初始化：启动两个网页爬取解析线程，将种子网页 1、2 分别加入到他们待爬取的栈中；

第一轮：种子网页 1 的网页爬取解析线程（下面简称线程 1）将栈中唯一的元素网页 1 出栈，任务调度线程发现该网页未被爬取，于是将其分配给了线程 1，解析网页 1 发现其链接了网页 2、4、3，将它们依次入栈，此时线程 1 的栈变成了 $[2,4,3]$；种子网页 2 的网页爬取解析线程（下面简称线程 2）将栈中唯一的元素网页 2 出栈，任务调度线程发现该网页未被爬取，于是将其分配给了线程 2，解析网页 2 发现其链接了网页 5、4，将它们依次入栈，此时线程 2 的栈变成了 $[5,4]$；

第二轮：线程 1 这边出栈网页 3，任务调度线程发现其未被爬取，于是分配给线程 1 进行爬取并解析网页 3，将它所链接的网页 6 入栈，此时线程 1 的栈变成了 $[2,4,6]$；线程 2 这边出栈网页 4，任务调度线程发现其未被爬取，于是分配给线程 2 进行爬取并解析网页 4，将它所链接的网页 6 入栈，此时线程 2 的栈变成了 $[5,6]$；

第三轮：线程 1 这边出栈网页 6，任务调度线程发现其未被爬取，于是分配给线程 1 进行爬取并解析网页 6，发现它没有链接网页入栈，此时线程 1 的栈变成了 $[2,4]$；线程 2 这边出栈网页 6，任务调度线程发现其已被爬取，则继续出栈网页 5，任务调度线程发现其未被爬取，于是分配给线程 2 进行爬取并解析网页 5，发现它没有链接网页入栈，此时线程 2 的空了，结束线程 2；

第四轮：线程 1 这边出栈网页 4，任务调度线程发现其已被爬取，则继续出栈网页 2，任务调度线程发现其也已被爬取。此时线程 1 的栈空了，结束线程 1。完成整个爬取过程。

【输入格式】

从标准输入读入数据。

第 1 行包括两个整数 M, N。M 表示整个网页文档集合的大小，N 表示初始种子 URL 集合的大小。

第 $2 \sim N+1$ 行每行包含一个网页的 URL，这 N 个 URL 构成初始种子 URL 集合。

第 $N+2 \sim M+N+1$ 行每行包含一个网页的 URL 以及网页中所包含的所有链接，两者中间用英文逗号隔开，网页中所包含的链接之间由空格分隔开，这些 URL 的处理顺序应与其在此处的顺序保持一致。

数据保证 URL 只由阿拉伯数字、大小写英文字母组成，且不重复。同时，网页中出现的链接都在网页文档集合当中，网页 URL 长度不超过 8 个字符。

【输出格式】

输出到标准输出。

输出 N 行，第 $k(1 \le k \le N)$ 行表示由第 k 个种子网页发起的网页爬取解析线程依次爬取到的网页 URL，URL 与 URL 之间用英文逗号隔开。

【样例 1 输入】

```
6 2
url1
url2
url1,url2 url4 url3
url2,url5 url4
url3,url6
url4,url6
url5,
url6,
```

【样例 1 输出】

```
url1,url3,url6
url2,url4,url5
```

【样例 1 解释】

样例中给出的六个网页之间的链接关系，以及调度方式，与【题目背景】中所提及的例子是一致的，因此线程 1 依次爬取了 `url1`、`url3`、`url6` 三个网页，线程 2 依次爬取了 `url2`、`url4`、`url5` 三个网页。

【子任务】

测试点编号	M	N
1	20	2
2	1000	5
3	10000	10
4	10000	20

任务 3：BM25 计算（19 分）

【题目背景】

BM25 算法作为一种计算文档和查询词之间相似度的方式，因其低廉的计算成本、不错的性能效果，自提出以来就得到了广泛的应用。请阅读《学习手册》中与 BM25 计算相关的内容，在给定文档集合的情况下，计算给定的文档和查询词之间的 BM25 指标。我们约定算法中的参数为 $k_1 = 1.2$，$k_3 = 2$，$b = 0.75$。

【输入格式】

从标准输入读入数据。

第 1 行包括两个整数 N, M。N 表示整个文档集合的大小，M 表示计算给定的文档和查询词之间 BM25 指标任务的数量。

第 $2 \sim N+1$ 行每行给出一个文档的内容，数据保证文档内容只由小写英文字母和空格组成，不同英文单词之间以空格隔开。数据保证文档内容不超过 5000 个字符。

第 $N+2 \sim M+N+1$ 行每行包含一个文档的编号（前面列出的 N 个文档从前到后编号分别为 $\{1, 2, 3, ..., N\}$）和查询词，两者中间用英文逗号隔开。查询词由一至多个英文字母组成。

【输出格式】

输出到标准输出。

输出 M 行，第 $k(1 \leq k \leq M)$ 行表示第 k 个 BM25 计算任务中计算得到的 BM25 值。输出结果保留 4 位小数，考虑到浮点数运算可能存在的误差，精度要求在标准答案 ± 0.001 区间之内即可。

【样例 1 输入】

```
6 3
english politics
english english
chinese physics chemistry english english math
history chemistry
english physics english chinese politics english biology chinese
   chemistry english
chinese english chemistry chemistry chemistry chemistry
4,chemistry
4,physics math chemistry
```

```
6,physics chinese
```

【样例 1 输出】

```
-0.7671
-0.7671
0.0000
```

【子任务】

测试点编号	N	M
1	50	5
2	50	50
3	1000	20
4	1000	1000

任务 4：PageRank 计算（22 分）

【题目背景】

PageRank 算法作为基于网页间的链接结构分析进行网页排名的著名算法，以往常被用来当做网页排序的重要指标。请阅读《学习手册》中 PageRank 算法相关部分，在给定的网页集合以及网页间链接结果关系下，采用迭代法计算各网页的 PageRank 值。迭代初始化各个节点的 PR 值设置为 $1/N$，N 为网页集合的网页个数，迭代收敛条件设置为对任意网页 A 都满足 $|PR'(A)-PR(A)| < 1.0 \times 10^{-6}$，阻尼系数 d 设置为 0.9。

【输入格式】

从标准输入读入数据。

第 1 行包括一个整数 N, 表示整个网页文档集合的大小。

第 $2 \sim N+1$ 行每行包含 N 个数（数取值均为 0 或 1），第 i 行的第 j 个数如果是 0 表示第 i 个网页没有指向第 j 个网页的链接，如果是 1 则表示第 i 个网页有指向第 j 个网页的链接。一行的 N 个数之间由空格隔开。

【输出格式】

输出到标准输出。

输出 N 行，第 $k(1 \le k \le N)$ 行表示第 k 个网页的 PageRank 值。输出结果保留 6 位小数，考虑到浮点数运算可能存在的误差，精度要求在标准答案 ± 0.00001 区间之内即可。

【样例 1 输入】

```
6
0 1 1 1 1 0
0 0 1 1 1 0
1 1 0 0 1 1
1 1 1 0 1 0
1 1 1 1 0 1
1 0 0 0 0 0
```

【样例 1 输出】

```
0.209356
0.173292
```

```
0.183902
0.150124
0.190915
0.092409
```

【子任务】

M 表示所有 N 个网页中包含的总链接数上限。

测试点编号	N	M
1	20	400
2	100	2000
3	500	8000
4	1000	10000

任务 5：基于决策树的文本分类（26 分）

【题目背景】

文本分类任务是信息检索和自然语言处理领域一个重要的研究内容。在有一定量的数据作为训练集合训练分类模型的基础上，对于任一段给定的文本内容，如何尽可能准确地将其分类准确，也一直受到研究者的广泛关注。

请阅读《学习手册》中文本分类以及决策树的相关内容，请以给定的训练文档数据集为基础（必须使用全部的训练数据，以确保模型与答案的标准模型一致），训练得到一个决策树分类模型，并基于此，对需要判别类别的一些文档内容进行分类。注意，在我们的决策树模型生成过程当中，只有当**所有特征都使用完，或者某一个分支下全部都是单一类别的文档时**，才终止当前节点的进一步剖分，否则我们希望这棵树分的越细越好。

另外，我们所采用的的分类特征均为某个单词是否出现（只考虑出现/不出现这两种类别，不对出现次数做进一步的特征使用）。由于训练样本中可能涉及到的单词数量非常多，整个搜索空间会非常大，因此在数据中我们也会给定一个特征单词的范围，仅仅需要以这些单词出现与否作为特征进行决策树的生成。

为了保证结果的唯一性，还约定：

- 在决策树建立过程中，若有多个不同的候选特征单词带来的信息增益（Information Gain）相同，则选取在特征单词列表中顺序最靠前的特征单词作为特征。
- 在决策树的某叶子节点上，若仍有两个类别无法区分（即在训练数据当中，所包含的这两个类别的文档数量相同并同为最多时），则选取分类标签编号较小的类别作为该叶子节点的分类类别。

【输入格式】

从标准输入读入数据。

第 1 行包括四个整数 N, M, L, K。N 表示整个训练文档数据集的规模，M 表示需要判别分类的文档个数，L 表示用作决策使用的候选特征单词的数量，K 表示待选的分类标签种类。

第 $2 \sim N+1$ 行数据为训练文档数据集。每行包含一个分类标签（整型数字，在集合 $\{1, 2, ..., K\}$ 中选取）以及文档的内容，两者之间用逗号隔开。文档内容由若干个英文单词组成（已完成文本预处理，单词只由 26 个小写英文字母组成），单词之间用单个空格隔开。

第 $N+2 \sim N+M+1$ 行每行包括一个待判别分类的文档内容，内容格式与训练集一致。

第 $N+M+2$ 行包含 L 个单词，表示用作决策使用的候选特征单词集合，单词之间由单个空格隔开。

【输出格式】

输出到标准输出。

输出 M 行，第 k $(1 \leq k \leq M)$ 行表示第 k 个待判别分类的文档内容的分类信息，包括一个分类标签以及该文档在决策树上使用的所有特征单词，两者之间由英文逗号隔开。分类特征单词之间由空格隔开，按照分类时决策树树根到分类所采用的叶子节点的顺序输出。

【样例 1 输入】

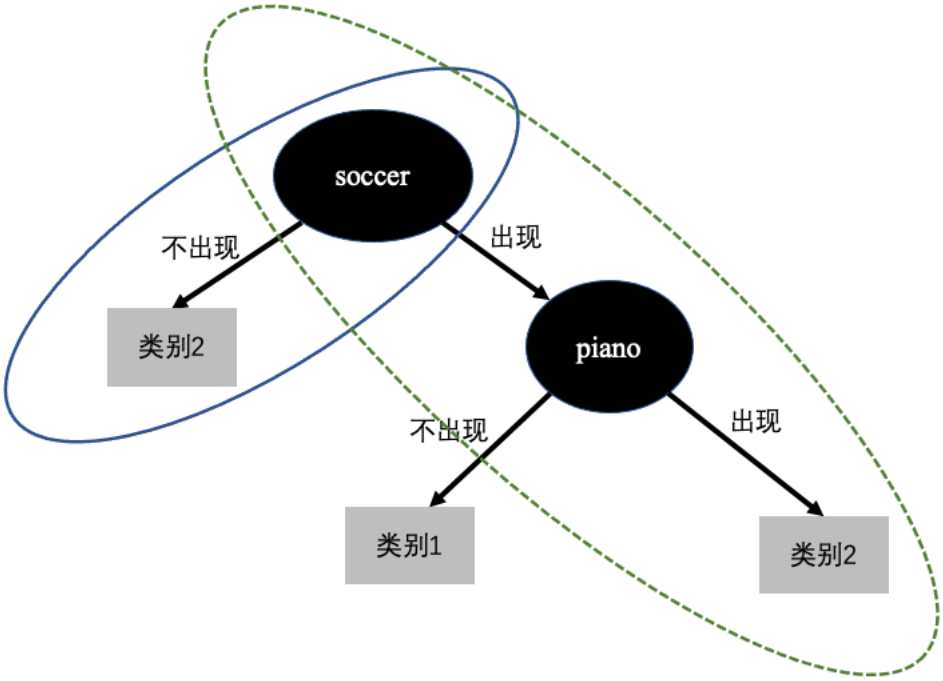
```
6 2 4 2
2,soccer for piano song
2,basketball is song
2,song
1,soccer with song
2,basketball as song
1,soccer
basketball
soccer piano song
soccer song basketball piano
```

【样例 1 输出】

```
2,soccer
2,soccer piano
```

【样例 1 解释】

下图给出了基于样例中的训练文档数据集所构建的决策树。黑色背景圆框表示分类节点，判定时根据文档内容下沉到子节点。灰色背景方框表示决策节点，根据该节点类别确定待分类文档所属类别。蓝色实心框、绿色虚线框分别表示第一、第二个待判定分类的文档内容的分类决策路径。



【子任务】

测试点编号	N	M	L	K
1	100	10	4	2
2	500	100	5	2
3	3000	100	6	2
4	10000	200	6	3