# 4.1排序算法

1、排序算法是最基本的算法，很多复杂算法都是以排序为基础进行构造的。关于排序算法，下列说法不正确的是\_\_\_\_\_。

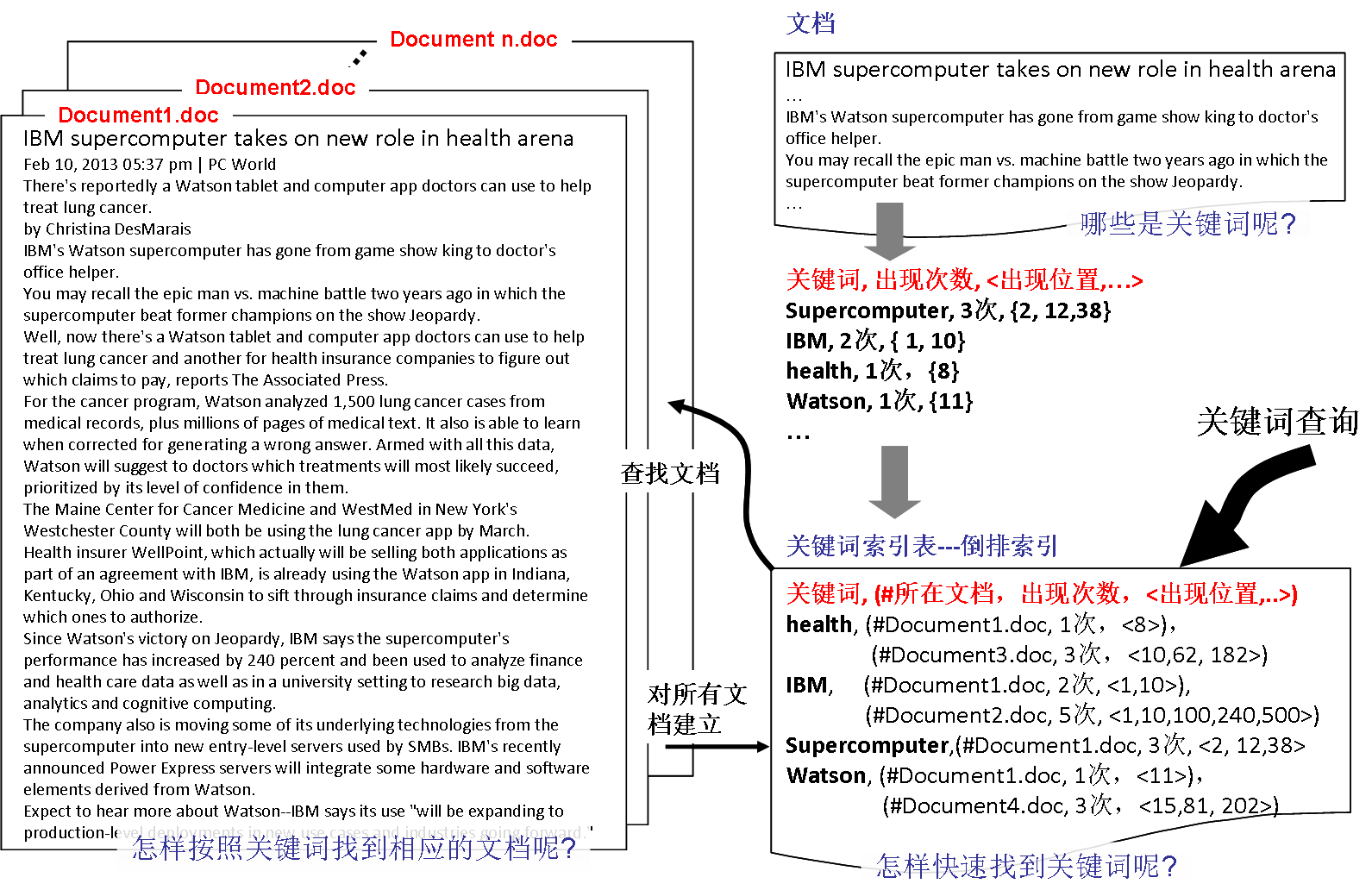
(A)大规模数据集合中查找有无某些元素的问题，有序数据集合比无序数据集合的查找要快得多；

(B)大规模数据集合中按元素分组进行计算的问题，有序数据集合比无序数据集合的计算要快得多；

(C)对无序数据集合，两个算法 X和Y：X采用无序数据处理，Y采用先将无序数据排序成有序数据，然后进行处理；则对前述(A)、(B)两类问题，Y算法一定比X算法慢；

(D)上述说法有不正确的；

3、关于“非结构化数据(文档)的查找与搜索”问题，参考下图，回答下列问题。注意每份文档可能包含数千数万的词汇。



(1)若要在n个全文文档中(n可能很大)查找有无某个关键词的文档，为提高检索效率，最好的做法是\_\_\_\_\_。

(A)直接用给定关键词来匹配每一份文档中的每一个词汇。若该文档存在匹配成功的词汇，则输出该文档；否则，不输出该文档。

(B)对这n个文档，首先建立一个“关键词”索引表，该索引表记录着“关键词”及包含该关键词的“文档编号”。在此基础上，用给定关键词来匹配索引表中的关键词。如果匹配成功，则输出索引表中相对应的文档编号；否则，则输出信息“没有含该关键词的文档”。

(C)对这n个文档，首先建立一个“关键词”索引表，该索引表记录着“关键词”及包含该关键词的“文档编号”，并按关键词进行字母序的排序。在此基础上，用给定关键词来匹配索引表中的关键词。如果匹配成功，则输出索引表中相对应的文档编号，否则，则输出信息“没有含该关键词的文档”。

(D)选项(B)(C)比选项(A)的做法好，但选项(B)(C)没有效率上的差别。

(3)针对下列问题求解方法：对n个文档，首先建立一个“关键词”索引表，该索引表记录着“关键词”，包含该关键词的“文档编号”，以及该关键词在该文档中出现的“次数”；对索引表，按关键词进行字母序的排序；如果关键词相同，则进一步按“次数”对同一关键词的若干文档进行降序排序。在此基础上，用给定关键词来匹配索引表中的关键词。如果匹配成功，则进一步寻找次数最多的m个索引项，输出相对应的文档编号；否则，则输出信息“没有含该关键词的文档”。问该方法涉及到几类算法，说法正确的是\_\_\_\_\_。

(A)涉及字符串的字母序排序算法；

(B)涉及数值属性排序算法；

(C)涉及字符串匹配算法；

(D)涉及数值属性查找算法；

(E)涉及上述全部算法；

5、下列三种算法是经常应用的内排序算法：插入排序、选择排序和冒泡排序。阅读下列算法，回答下列问题。

**INSERTION-SORT(A)**

***1. for i=2 to N***

***2. { key = A[i] ;***

***3. j =i-1;***

***4. While (j>0 and A[j]>key) do***

***5. { A[j+1]=A[j];***

***6. j=j-1; }***

***7. A[j+1]=key;***

***8. }***

***SELECTION-SORT(A)***

***1. for i=1 to N-1***

***2. { k=i;***

***3． for j=i+1 to N***

***4. { if A[j]<A[k] then k=j; }***

***5. if k<>i then***

***6. {***

***7. temp =A[k];***

***8. A[k]=A[i];***

***9. A[i]=temp;***

***10. }***

***11. }***

***BUBBLE-SORT(A)***

***1. for i=1 to N-1***

***2. { haschange=false;***

***3. for j=1 to N-i***

***4. { if A[j]>A[j+1] then***

***5. { temp =A[j];***

***6. A[j]=A[j+1];***

***7. A[j+1]=temp;***

***8. haschange=true;***

***9. }***

***10. }***

***11. if (haschange ==false) then break;***

***12. }***

(1)关于INSERTION-SORT算法的基本思想，下列说法正确的是\_\_\_\_\_。

(A)一个元素一个元素的处理。每次处理一个元素，通过与当前已排序元素的比较，将该元素放入到当前正确排序的位置。直到最后一个元素则算法结束。

(B)一个轮次一个轮次的处理。将元素集合分成两个部分，已排序元素集合和未排序元素集合，开始时已排序元素集合为空。在每一轮次，从未排序元素集合中找出最小值的元素，将其移入已排序元素集合；直到未排序元素集合为空时则算法结束。

(C)一个轮次一个轮次的处理。在每一轮次中依次对待排序数组元素中相邻的两个元素进行比较：如不符合排序关系，则交换两个元素。直到某一轮次没有元素交换发生则结束。

(D)上述说法都不正确。

(3)关于BUBBLE-SORT算法的基本思想，下列说法正确的是\_\_\_\_\_。

(A)一个元素一个元素的处理。每次处理一个元素，通过与当前已排序元素的比较，将该元素放入到当前正确排序的位置。直到最后一个元素则算法结束。

(B)一个轮次一个轮次的处理。将元素集合分成两个部分，已排序元素集合和未排序元素集合，开始时已排序元素集合为空。在每一轮次，从未排序元素集合中找出最小值的元素，将其移入已排序元素集合；直到未排序元素集合为空时则算法结束。

(C)一个轮次一个轮次的处理。在每一轮次中依次对待排序数组元素中相邻的两个元素进行比较：如不符合排序关系，则交换两个元素。直到某一轮次没有元素交换发生则结束。

(D)上述说法都不正确。

(5)关于三种排序算法，下列说法正确的是\_\_\_\_\_。

(A)三种算法的时间复杂度都为O(n2)，所以三种算法的执行效率是一样的；

(B)尽管三种算法的时间复杂度都为O(n2)，但细致比较还是有差别的，例如冒泡法排序比选择法排序要快一些；

(C)尽管细致比较三种算法的执行时间是有差别的，但这种差别对内排序问题而言是可以忽略不计的；

(D)尽管细致比较三种算法的执行时间是有差别的，这种差别对内排序问题而言是重要的，因为内排序算法可能要被频繁的执行。

(7)阅读SELECTION-SORT算法，关于第3.行至第4.行间程序段的作用，下列说法正确的是\_\_\_\_\_。

(A)循环地在未排序元素集合中找最小值元素的位置，该位置保存在变量k中；

(B)循环地在未排序元素集合中找最小值元素，该元素保存在变量k中；

(C)循环地在未排序元素集合中找最大值元素的位置，该位置保存在变量k中；

(D)循环地在未排序元素集合中找最大值元素，该元素保存在变量k中；

(9)阅读BUBBLE-SORT算法，下列说法正确的是\_\_\_\_\_。

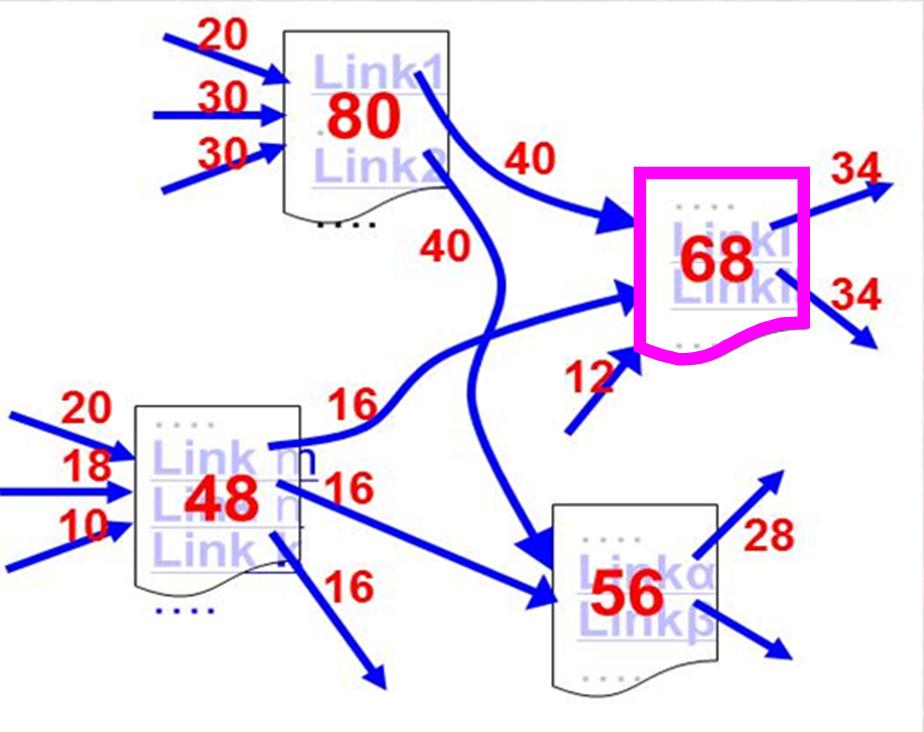
(A)该算法在N=20时，必定要执行20个轮次的内循环；

(B)该算法在N=20时，必定要执行19个轮次的内循环；

(C)该算法在N=20时，最多要执行20个轮次的内循环；

(D)该算法在N=20时，最多要执行19个轮次的内循环；

7、PageRank是Google公司提出的计算网页重要度的一种方法。参见下图，简单而言，网页是由“文本”和“链接”构成的，“链接”可使用户从一个网页跳转到另一个网页。因此，所谓“链接”即是某一个网页的地址，通过网页链接的读取，可以建立起各个网页之间的链接关系。对一个网页而言，其链接到其他网页的链接被称为“正向链接”，而所有链接到该网页的链接被称为“反向链接”。关于PageRank算法，回答下列问题。



图I.

(1)关于PageRank计算网页重要度的基本思想，下列说法正确的是\_\_\_\_\_。

(A)反向链接数越多的网页越重要----被链接次数越多越重要；

(B)反向链接加权和越高的网页越重要----被重要网页链接次数越多越重要；

(C)正向链接数越多的网页，其链接的权值越低----正向链接数越多的网页越不重要；

(D)上述全部。

(3)按照PageRank的思想，一个网页链接的权值被定义为\_\_\_\_\_。

(A)网页重要度除以该网页所拥有的正向链接数；

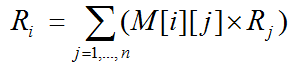
(B)网页重要度除以该网页所拥有的反向链接数；

(C)网页重要度除以该网页所拥有的所有链接数；

(D)上述都不正确；。

(5)PageRank算法中出现了一个“转移概率矩阵”，参见下图，其意义是\_\_\_\_\_。

(A)转移概率矩阵是基于网页链接关系矩阵AT计算得到的，按列来看是，网页j有多少个正向链接，其权值为多少分之一，反映了链接权值的计算方法；

(B)转移概率矩阵是基于网页链接关系矩阵AT计算得到的，按行来看是，网页i有多少个反向链接及其权值，可反映网页i的重要度计算方法即：，由其他网页的重要度及其权值计算该网页i的重要度；

(C)网页i的重要度Ri可以迭代地计算得到，设第m次得到的Ri记为Ri(m)，称为网页重要度Ri的一个状态，则状态转移概率为Ri由一个状态转变为另一个状态的概率；

(D)状态转移概率可广泛用于计算客观事物呈现状态序列S(0),...., S(m-1),S(m),...，而S(m)的计算仅由S(m-1)的值来确定的情况；

(E)上述说法都正确。

(7)前述说过 PageRank，通过不断地计算R(m)=M×R(m-1)来计算网页重要度，即由第(m-1)次的网页重要度来计算第(m)次的网页重要度，那么网页重要度的初始值R(0)应如何获得呢?

下列说法正确的是\_\_\_\_\_。

(A)随机产生各网页重要度的一组值，该组值对最终计算结果没有影响；

(B)由专家给出各网页重要度的一组值，该组值的质量好坏直接影响计算结果；

(C)设定各网页重要度都是1；

(D)随机产生各网页重要度的一组值，使网页重要度界于0和1之间，但该组值对最终结果没有影响。