*A.T.M. Shafiqul Khalid (Tuhin)

Ranking Problems in Practical Life

Introduction:

Ranking is an interesting problem in our practical life. In a football tournament, cricket tournament etc. ranking is probably the most sensative issue. So, the ranking procedure should be generally acceptable. Point sharing or formal match worsly effect the tournament result in conversional ranking technique and sometimes good teams drop out the tournament. The problem also present in sensative tournament like world cup. Therefore, Any ranking procedury will be more acceptable if it minimizes the effect of those illicit policy ranking has wide area of application. This technigue can also be used in project selection, product selection, music ranking etc.

A typical ranking problem

We will discuss some common problems related to ranking with some remedy. Let us consider a typical football tournament result. Team A, B, C, D, E and F partcipate in the tournament. Their result is as follows :

Table 1:

Team	Result
A-B	0-1
A-C	1-0
A-D	1-0
A-E	1-0
A-F	1-0
B-C	0-1
B-D	1-0
B-E	1-0
B-F	1-0
C-D	0-1
C-E	0-1
C-F	1-0
D-E	0-1
D-F	1-0
E-F	0-1

Table-2

Team	Total win	Goal difference	Against	For	Points
A	4	3	1	4	6
B	4	3	1	4	6
C	2	-1	3	2	4
D	2	-1	3	2	4
E	2	-1	3	2	4
F	1	-3	4	1	2

For each win 2 points have been assigned. The question is which one is the champion team, both A and B have same points 6. You may propose for an extra match between them. But if there is more than two teams then what is the solution. Probably you have to arrange a mini tournament. Let A be the champion and B be the 1st runners up then who is the 2nd runners up C, D or E. Team C,D and E all get 4 points so to find 2nd runners you must have to choose an acceptable criterion. Mini tournament among them may be a probable solution but not applicable always. For example out of 12 teams 6 teams may get same points. To rank the 6 teams mini tournament is arranged among them and 3 of them get same point and this process may continue upto several steps. However it is possible to give an acceptable result from the above table. B should be declared as champion, not A because B defeat A, although both get some points. Comparative study between A and B states B is better. Similarly among C, D and E , E is the best, D is 2nd and then C because E defeats both C, and D and D defeats C. Therefore the sequence should be B, A, E, D, C and F which is more acceptable and satisfies all of them.

Other Similar Type of Problems :

Do you ever think about ranking procedure of some complicated things to evalute like music. You feel better to answer the question "Is music M1 is better than music M2?" than ranking several music at a time. Your answer against the question for each set of pair is nothing but a tournament result. For example your answer for 6 music M1, M2, M3, M4, M5, and M6 may be as follows:

Table 3

Question	Answers
Is M1 better than M2?	No
Is M1 better than M3?	Yes
Is M1 better than M4?	Yes
Is M1 better than M5?	Yes
Is M1 better than M6?	Yes
Is M2 better than M3?	No
Is M2 better than M4?	Yes
Is M2 better than M5?	Yes
Is M2 better than M6?	Yes
Is M3 better than M4?	No
Is M3 better than M5?	No
Is M3 better than M6?	Yes
Is M4 better than M5?	No
Is M4 better than M6?	Yes
Is M5 better than M6?	No

Is Table 3 Similar to Table 1? Each music can be considered as a team and question is a match between two team. Answer "Yes" is a match result of 1-0 and "No" is a match result of 0-1. Now if you can rank Table 1 then similar technique will be for Table 3.

Table 3 can be summarized as follows :

Table 4

Music	Number of Yes	Better than
M1	4	M3, M4, M5, M6
M2	4	M1, M4, M5, M6
M3	2	M2, M6
M4	2	M3, M6
M5	2	M2, M3
M6	1	M5

An MD of an industry can apply similar technique in choosing some iteams to produce from a set of iteams.

Solution of the problem in computer science :

We will now discuss how computer science deal with this type of problem of ranking in general. A set of solutions has been proposed in the literature since arising this problem. All of them assume that

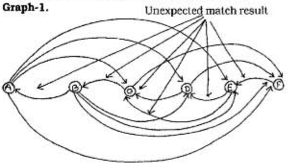
1. There is a natural ranking among the teams participating in the tournament.
2. If team A is better than team B in the natural ranking and B defeats A in practical tournament then the match will be considered as unexpected match (Error match also).
3. Any ranking will be considered as the natural ranking that gives minimum unexpected result.

Here natural ranking is the absolute ranking of the teams and to find out that each pair is asked to play a match. Our conventional ranking method does not guarantieed the absolute ranking. Therefore there is al-

* A.T.M. Shafiqul khalid is a final year under graduate student in the Dept. of Computer Science and Engineering, BUET.

ways a possibility of having better ranking in terms of minimal number of unexpected match. Previous ranking result can be graphically represented as follows :

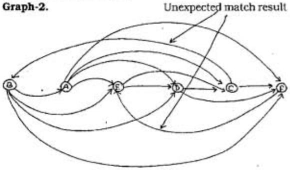
Graph-1.



Arc from A to C in the figure means that A defeats C. In the same way it can be stated that C is defeated by D and B defeats F. If we consider that ABCDEF is the natural ranking then 6 reverse arc implies that there is 6 unexpected match (the result should not be). If the result of those 6 matches would reverse then we would get the absolute ranking.

Now considered the following graph that represent the same tournament result.

Graph-2.



This graph has only 2 reverse arcs implying that if B A E D C F is considered as the absolute ranking and only 2 unexpected result outcome. Certainly BAEDCF is better ranking than ABCDEF. Now the problem is how the 2nd graph can be developed from the 1st graph. If number of team rises then the graph will be very complicated and a manual calculation will be of the same degree. Our proposed algorithm (procedure for ranking) in the paper already published was MST algorithm based on Majority spanning tree. Before our proposed MST Algorithm, Arrange and GIK was the most efficient available algorithm in the literatures. Statistical result proves that our MST is the best in most cases (above 80% cases) giving a ranking with minimum number of unexpected result. More over in 90% cases our proposed algorithm is able to improve the solution obtain from GIK and Arrange for large tournament size. For further information readers are asked to consult with the journal, Computer & OPS. Res., Pergamonpress, Vol 22, No. 2, pp. 221-226.

Conclusion

Table 5

Team	Win
A	4
B	5
C	1
D	2
E	3
F	0

If we just invert the match result (E,F) and (B,C) than the result will be as in table 5.

Now ranking BAEDCF can easily be obtained from the table. The problem is to define to develop a procedure that will generate suck straightforword result by inversting minimum match result. If we draw a graph from table 3 then similar graph like graph 1 results and certainly graph 2 will represent the acceptable graph and hence ranking of the music in table 3 will be M2, M1, M5, M4, M3, M6.

The MD can first rank the items and choose top teams. If the MD has to choose three project from 6 project P1, P2, P3, P4, P5 and P6 then he must first develop a table like table 3. through comparative study. It is very easy for him to compare only two project at a time. He will be confused with multiple projects. If his comparative stydy give similar result like table 3 then ranking of the project will be P2, P1, P5, P4, P3 and P6 and he must choose P2, P1 and P5. From the discussion it can also be statated that cenvensional Ranking method cannot resist formal game (where two teams came to an agreement about result before the match) or the game where two teams intensionaly share points. Proposed solution minimizing unexpected result will not only eliminate these type of illegal mentality of the teams but also give some criteria of ranking where lottery is a must . If you have any comment or suggestion improving ranking policy, please write or contact me to the following address.

Tuhin
Shahid Smriti Hall-2414
BUET

Appendix

MST Algorithm can be summerized as follows :

```
for i= 1 to Tournament Size -1
    for j = i + 1 to Tournament Size do
        for k = i to j - 1 do
            if swapping of set (i, k) and (k+1,j)
             give improvement then
              swap the set and restart from i-loop.
```

[This article is based on newly published (February, 1995) research paper titled "A New Algorithm for Ranking Player of a Round Robin Tournament" in the journal Computers & Operations Research (COR), Pargamon press, Great Britain. Other members of this research were, key person Dr. M. Kaykobad, Q.N.U Ahmed and Rezwan-AL-Bakhtiar.]

BUSINESS INFORMATION SYSTEM AND DATABASE APPROACH

Md. Arif Hasan

We are now in an information age, one in which the management of the information resources of a corporation will be vital importance. Business information system are systems that use these resources to convert data into information in order to improve productivity. An example of a Business Information System is retail store system that converts sales transaction data into information used in the preparation of customer billings, inventory management and calculation of profit and loss. Computer is an important resource that support the business information system. Historically, computer prices were very high and the computer was the exclusive domain of the big company. So, we have relied on people to collect, edit and disseminate information. But an excessive number of people have become a resource that many business cannot afford. Further, there are innumerable business that are using people for information processing and do not yet realize that they cannot afford the combination. But now time has changed and such is not the case anymore. Computer prices are coming down. Equally important, new microcomputer are becoming widely available. Based on these miniature components, the small business manager has an opportunity to improve performance on several fronts. He or she can reduce dependence on labour to perform routine tasks. Much of the arduous bookkeeping and record keeping can be done via automation. The fundamental reason why a microcomputer is acceptable for the tasks of a small but comprehensive, integrated information system is the concept of database design. To understand the full impact and importance of this database approach, it is probably important to compare this database approach with the traditional approach which have been used historically.

DATABASE VS TRADITIONAL APPROACH :

Historically, computer did one task at a time because they had small memories and access to data storage was slow and cumbersome. Thus, a structure like *Figure 1* developed. Small data storage files were developed to support each application. For example, the accounts payable data files supported the accounts payable application program and so on. Sometimes applications shared a common data files (see bold line in figure 1), but these were usually evolutionary changes in the programming structure. As new applications were needed, new files were developed . The result was that if a consumer changed his or her mailing address, the change might be required on half a dozen files, where that customer's name was carried as a record element. Hence, file maintenance becomes but expensive and difficult. As files grew larger they generally became less accurate.

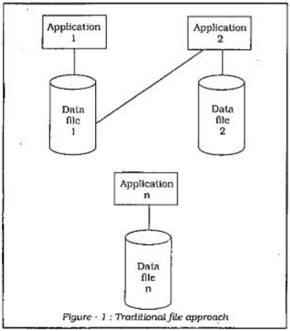


Figure - 1 : Traditional file approach

On the other hand, the database approach was developed s a natural and necessary evolutionary step. Conceptually, it is as shown in figure 2. All data are stored in a central filling system called *database* and all the applications access and modify the various elements in those database using master set of logic. This approach became a possibility when larger, faster microcomputers and quicker random-access storage system became technologically possible. The database concept became a necessity as the scope of the data processing activity became larger and more comprehensive, thus increasing the sizes of the files.

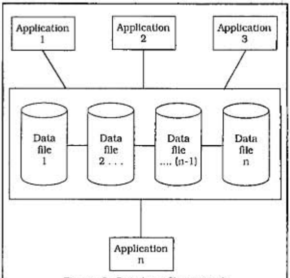


Figure - 2 : Data base file approach

In the database approach, the customer's address, for example, in stored in one place. If that address is changed, only one input is required to the system, as all the various files in the database are interlocked. For example, the file containing merchandise items also includes the number of vendor who supplies that item. The item is automatically linked through that vendor number to a file and an address in that file which contains all information about the vendor. Using this method, one can use a minimum amount of storage area for maximum information storage and accessibility.

ADVANTAGE OF DATABASE

From the above comparison, we can find the following advantages to a database approach :

a. Processing time and the number of programs written are substantially reduced.
b. All applications share centralized files.
c. Storage space duplication is eliminated.
d. Data are stored once in the database and are easily accessible when needed.

DRAWBACKS OF DATABASE

The two drawbacks of database approach are the cost of specialized personnel and the need to protect sensitive data from unauthorized access. ❋

A Focus On Disks-DOS Perspective

K. A. M. Morshed

(Concluding Part)

The File Allocation Table (FAT)

As I mentioned before, FAT is used to chain together a record of where a file's data is stored. Here is how it works. The FAT is divided into fields that corresponds to ach of the assignable clusters on the disk. That is, the FAT contains a field for each of the assignable clusters. These field is either 12 or 16 bit long. If the total number of clusters exceed hex FFF or decimal 4087 then the field length is 16 bit, otherwise it is 12 bit. Whenever we create a file, the file entry in the directory contains a starting cluster number for it. This is the cluster where the file's data appears first in the data area. The corresponding cluster entry on the FAT is then marked as occupied. This is the entry point into the FAT for the file. As the file grows, more clusters are allocated to the file. Whenever the second cluster is allocated, the first cluster slot on the FAT contains the number of the second. Again, when the third is allocated, the second FAT slot for the file contains the number of the third. If we suppose that the file is just three cluster long, the slot of FAT corresponding to the third cluster allocated contains an end of the chain marker.

The above is the hex dump of the first 224 bytes of the FAT of the 80 MB hard drive we are examining. Let us look at the figure closely. The first two entry of the FAT is dummy. The first entry, that is FFF8 is simply a version of the original media descriptor F8. Normally, the second entry almost always contains FFFF. From the third entry the actual table starts. See, the third entry contains 0003, the number of the fourth cluster, and the fourth entry contains 0004, the number of the fifth cluster and so on. That is, the file which has occupied the third cluster is also spread through fourth and fifth cluster. Actually, the end of chain mark at the 22nd entry tells us that the file is ended at 22nd cluster. Given the cluster size, here it is 2048 Byte, we can say that the file has occupied 44 kilobytes of space on the disk.

Before ending our discussion on FAT let us see, what a FAT entry actually means. Just browse through the following table. The first character in the value field is enclosed inside a pair of brackets. This is done because if the FAT is a 16 bit one then all the four digit is significant while if it a 12 bit one then the first character enclosed inside the brackets is simply isn't there. As you can see, if FAT entry contain (F)FF7 then may be it indicates a bad cluster. We lay emphasis on the word may be because, the above is true only if (F)FF7 is not a normal entry to the next cluster of the file. So, to be sure, you have to go through the chains to see whether it is a normal entry or a bad cluster mark.

The Root directory

The root directory like any directory of any disk contains information related to files. The only difference is, this is the default directory, and technically ultimate parent of all the sub-directories. Not only that, root directory requires special attention also because it has a specific length which is determined on the basis of disk type. You see, as the disk has 2 FATs and each occupied 170 sectors and there on boot sector, the root directory starts at 342nd sector or at sector number

Value	Meaning
(0)000	Cluster available
(F)FF0 -(F)FF6	Reserved cluster
(F)FF7	Bad cluster, if not used
(F)FF8 - (F)FFF	Last cluster of the fill
Other	Next cluster in the chain

Figure 4 M : eaning of FAT entry

8 BIT hex dump of the first 224 Bytes of a 80 MB disk.

```
0200  F8  FF  FF  FF  03  00  04  00-05  00  06  00  07  00  08  00  .........
0210  09  00  0A  00  0B  00  0C  00-OD  00  0E  00  0F  00  10  00  ..........
0220  11  00  12  00  13  00  14  00-15  00  FF  FF  17  00  18  00  .........
0230  19  00  1A  00  1B  00  1C  00-1D  00  1E  00  1F  00  20  00  .........
0240  21  00  22  00  23  00  24  00-25  00  26  00  27  00  28  00  !.".#.$.%.&.'.(.
0250  FF  FF  2A  00  2B  00  2C  00-2D  00  2E  00  2F  00  30  00  ..*.+.,.-.../.0.
0260  31  00  32  00  33  00  34  00-35  00  36  00  37  00  38  00  1.2.3.4.5.6.7.8.
0270  39  00  3A  00  3B  00  3C  00-3D  00  B4  00  FF  FF  40  00  9.:.;.<.=....@.
0280  41  00  FF  FF  43  00  44  00-45  00  FF  FF  47  00  48  00  A...C.D.E...G.H.'
0290  0A  8F  4A  00  FF  FF  4C  00-FF  FF  4E  00  FF  FF  50  00  ..J...L...N...P.
02A0  FF  FF  52  00  FF  FF  54  00-FF  FF  56  00  FF  FF  58  00  ..R...T...V...X.
02B0  FF  FF  5A  00  FF  FF  5C  00-FF  FF  FF  FF  FF  FF  FF  FF  ..Z...\........
02C0  FF  FF  64  00  FF  FF  FF  FF-7C  00  FF  FF  FF  FF  FF  FF  ..d......|........
02D0  FF  FF  FF  FF  FF  FF  FF  FF-FF  FF  FF  FF  FF  FF  FF
```

16 BIT equivalent of the above FAT

```
FFF8  FFFF  0003  0004  0005  0006  0007  0008  0009  000A  000B  000C  000   D000E
000F  0010  0011  0012  0013  0014  0015  FFFF  9917  0018  0019  001A  001B  001C
001D  001E  001F  0020  0021  0022  0023  0024  0025  0026  0027  0028  FFFF  002A
002B  002C  002D  002E  002F  0030  0031  0032  0033  0034  0035  0036  0037  0038
0039  003A  003B  003C  003D  00BA  FFFF  0040  0041  FFFF  0043  0044  0045  FFFF
0047  0048  8F0A  004A  FFFF  004C  FFFF  004E  FFFF  0050  FFFF  0052  FFFF  0054
FFFF  0056  FFFF  0058  FFFF  005A  FFFF  005C  FFFF  FFFF  FFFF  FFFF  FFFF  0064
FFFF  FFFF  007C  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF  FFFF
```

3 FAT entry of the disk

341. From our discussion on BIOS parameter block, we have learned that the hard disk we are examining is capable of storing information related to 512 files. It takes 32 Bytes of space to store a single file's information. So, it takes 16 Kilo Bytes of space to store the entire root directory. The following figure depicts the root directory of the disk we are examining so far.

1. File name (8 Bytes): The first eight bytes of a root directory entry contains the name of the file.
2. Extension (3 Bytes): These bytes of a root directory entry contains the file name extension. Since, the disk is a bootable one, the first file is IO.SYS.
3. Attribute (1 Byte): This byte contains the file's attribute. See text for details.
4. Reserved (10 Bytes): This space is reserved for future use.
5. Time stamp (2 Bytes): This word contains the time when the file was created or last updated. See text for detail.
6. Date stamp (2 Bytes): This word contains the date when the file was created or last updated. See text for details.
7. Starting cluster (2 Bytes): This word stores the starting cluster number of the file. This is the number of cluster where the file's data first appears.
8. File size (4 Bytes): This double word contains a integer representing the file's actual size in bytes.
9. The deletion marking: The first character of a file entry contains E5 only if it was deleted. However all the other information remains intact unless the file is overwritten to accommodate new files.

A. Sub-directory attribute: The attribute of hex 10 or decimal 16 indicates that the corresponding entry is a sub-directory.

B. Normal archive file: The attribute of hex 20 or decimal 32 indicates normal file with archive BIT on.

Root directory

```
0000  49   4F  20  20  20  20   20  20_1 53  59   53_2 27_3 00  00  00  00   IO      SYS'....
0010  00   00  00  00  00  00_4 00  30_5 4C  1A_6 02   00_7 16  9E  00  00_8 ....... 0L.......
0020  4D   53  44  4F  53  20   20  20   53  59   53   27   00  00  00  00   MSDOS   SYS'....
0030  00   00  00  00  00  00   00  30   4C  1A   16   00   FA  94  00  00   .......0L.......
0040  E5_9 48  4B  4C  49  53   54  20   43  50   53   20   00  00  00  00   .HKLIST CPS ....
0050  00   00  00  00  00  00   31  64   39  1B   65   00   36  00  00  00   ......1d9.e.6...
0060  43   50  41  56  20  20   20  20   20  20   20   10_A 00  00  00  00   CPAV       .....
0070  00   00  00  00  00  00   E4  A6   95  1A   9F   00   00  00  00  00   ................
0080  54   4F  44  41  59  20   20  20   45  58   45   20_B 00  00  00  00   TODAY   EXE ....
0090  00   00  00  00  00  00   00  60   59  0E   C0   A7   9C  04  00  00   .......'Y.......
```

5 Hex dump of Root directory

Now let us explore the root directory entries in a bit detail. The first Byte of each root directory entry can have one of the following values.

Value (Hex)	Meaning
00	Directory entry has never been used
05	First character is actually E5
2E	Entry is an alias for current directory: if followed by another 2E then it is the alias for parent directory
E5	File has been erased

As you can see, if we delete a file the first character of the name field becomes ASCII E5 Hex. But if we want to start a file name with E5 then the root directory will contain 05 instead. The attribute Byte of a file entry is mapped using the coding presented in the table.

Coding Scheme of Attribute Byte

BIT	Meaning	BIT	Meaning
0	Read-only	4	Directory
1	Hidden	5	Archive
2	System	6	Reserved
3	Volume label	7	Reserved

The Attribute byte of the first file that is IO.SYS is hex 27 or Binary 00100111. That is, the file attribute is read-only, Hidden, System and Archive. The date and time fields of the file entry is coded using the following coding technique.

Coding Scheme for Date and Time

Date		Time	
BITs	Content	BITs	Content
0-4	Day of the month (1-31)	0-4	Seconds (2 sec. increment)
5-8	Month (1-12)	5-10	Minutes (0-59)
9-15	Year (Relative to 1980)	11-15	Hours (0-23)

The Date and Time stamp for IO.SYS file is 1A4C and 3000 respectively. The Binary equivalent of the Date stamp is 0001101001001100. If we divide this binary stamp into parts, then it becomes 01100 for day, 0010 for month and 0001100 for the year. Converting them back to decimal we get 12 for month, 2 for day and 13 for year. Remember the year figure is relative to 1980. That is the actual year becomes 1980+13=1993. Thus the date stamp becomes 12-2-93.

Likewise if we convert the time stamp into Binary then we get 0011000000000000. Dividing the number into parts we get 00110 for hour, 000000 for minutes and 000000 for seconds. That is the time becomes 6:00:00.

With this we like to finish this article. Hope, the above discussion will inspire you to learn more about disks. After all, that was the only intention of this article. ≠