

BASIC 활용을 위한 분류알고리즘의 비교 분석

강 성 모
서울 경기상업고등학교

(1987년 5월 20일 받음)

I. 서론

현대 사회는 급속도로 발전하는 과학 기술 문명에 힘입어 고도의 정보화 사회로 전환해 가고 있다. 컴퓨터는 이같은 현대 사회에서 없어서는 안될 존재로서 정부·기업·연구소·학교 등은 물론 가정의 개인에게까지 보급되어 각기 목적에 따라 활용되고 있다. 이러한 현상은 컴퓨터에 대한 올바른 인식의 습득을 필요로 하게 되었고, 교육이 시대적 조류에 부응하면서 국가 사회의 선도적 역할을 담당하여야 한다는 측면에서 볼 때 오늘날과 같은 시대적 요구에 대한 교육의 필요성과 합리적이고 효과적인 방향 설정에 따른 컴퓨터 교육의 실시는 필연적이라 아니할 수 없다.

우리 나라에서 컴퓨터 교육이 활성화된 계기는 1970년 7월 “고등학교 이상의 학교에서 컴퓨터 교육이 필요하다”는 대통령 지시에 의하여 문교부를 중심으로 관계 부처와 협의를 거쳐 수립된 「전자 계산기 교육 계획」이라 볼 수 있으며 이에 의하여 실업계 고등학교에서는 전자 계산 관련 교과목을 자도하였고,

또한 1981년 문교부 고시에 의하여 상업계 고등학교에 정보처리과를 두어 정보화 사회에 필요한 직업인을 양성할 수 있도록 교육 과정이 개편되어 1984년도부터 새 교육 과정에 의한 컴퓨터 교육이 실시되고 있다.

일반 교육으로서의 컴퓨터 교육은 1983년 문교부장관의 “국민학교부터 컴퓨터 교육을 실시한다”는 발표에 따라 일반 고등학교에서는 물론 중학교와 국민학교에서 선택 과목으로 컴퓨터 교육을 실시하거나 특별 활동 시간을 활용하여 적극적으로 교육되고 있는 실정이다.

따라서, 일반 교육으로서 컴퓨터 교육을 효과적으로 시행하기 위해서는 각급 학교의 컴퓨터 교육의 전반적인 실태 분석과 이에 따른 교육 목표 및 교육 내용의 설정, 교육 실시 방안과 교육에 필요한 시설 및 설비 요건, 그리고 학교 교육에서 컴퓨터의 활용을 촉진시키기 위한 각종 프로그램의 개발 등의 많은 연구가 필요하다 하겠다.

본 연구는 컴퓨터를 이용하여 자료 처리를 할 때 아주 자주 쓰이는 분류(Sorting) 알고리즘을 비교·분석

하여 봄으로써 문제를 분석하고 해결하는데 도움을 줄 뿐만 아니라 새로운 분류 알고리즘을 개발하는데에도 도움이 되도록 하는데 그 목적이 있다. 또한 자료 처리의 각 경우에 알맞은 알고리즘을 선택할 수 있도록 각 분류 프로그램의 분석 결과를 수록하였다.

일반적인 경우 알고리즘을 프로그램으로 표현할 경우 파스칼(PASCAL)과 같은 구조적 언어를 사용하는데, 본 논문에서는 구조적 언어를 접하지 못한 초·중·고생들이나 이를 지도하는 교사들이 쉽게 이해할 수 있도록 퍼스널 컴퓨터와 함께 널리 보급된 베이직(BASIC) 언어를 이용하여 기술하였다.

II. 분류(Sorting)

분류의 종류는 크게 두 가지로 구분할 수 있는데 첫째는 내부 분류(Internal Sorting)로서 주기억 장치내에서 수행할 수 있는 정도로 자료의 양이 적을 때 사용하는 분류 방법이고, 둘째는 외부 분류(External Sorting)로서 자료의 양이 많아서 자기디스크나 자기테이프와 같은 보조 기억 장치를 이용하여 분류하는 방법이다.

내부 분류와 외부 분류에 대한 종류는 다음과 같다.

1. 내부 분류(Internal Sorting)

(1) 비교 분류(Comparative sorting)

비교하고자 하는 각 레코드의 키(key)값을 한 번에 두 개씩 비교함으로써 분류하는 방법으로 다음과 같은 종류가 있다.

- ① Bubble Sorting
- ② Selection Sorting
- ③ Insertion Sorting
- ④ Shell Sorting
- ⑤ Heap Sorting
- ⑥ Quick Sorting

이 방법은 여분의 기억 장소를 필요로 하지 않는 특징이 있다.

(2) 분배 분류(Distribution Sorting)

비교하고자 하는 각 레코드의 키값을 한 번에 한 비트씩 검토하여 분류하는 방법으로 그 종류는 다음과 같다.

- ① Radix Sorting

- ② Radix exchange Sorting

분배 분류는 상당한 여분의 기억 장소를 요구하기도 한다.

- (3) 주소 변환 분류(Address calculation Sorting)

해싱(hashing) 기법을 이용하여 비교하고자 하는 키값을 번지로 변환시켜 분류시키는 방법을 말한다.

2. 외부 분류(External Sorting)

컴퓨터에 의한 자료 처리 과정에서 순서적으로 분류해야 할 자료들이 매우 많아서 주기억 장치를 이용하여 한 번에 내부 분류로서 모든 자료들을 분류시킬 수 없을 때 사용하는 방법으로 그 종류는 다음과 같다.

- ① Disk Sorting
- ② Tape Sorting

III. 분류 프로그램

1. 사용 컴퓨터와 주변 장치

본 연구에 사용된 컴퓨터는 16비트 IBM PC의 호환 기종인 효성 PC-16000으로 플로피 디스크 드라이브가 2대 부착되었으며 GREEN MONITOR를 사용하였고, 프린터는 효성 HPP-16B를 사용하였다.

현재는 16 비트 IBM PC의 호환 기종은 소수의 실업계 고등학교에 소량이 보급되어 있는 실정으로 일반 초·중·고교에서는 이용되기 어려운 실정이나 가까운 장래에 널리 보급되어 사용되리라 생각한다.

2. 사용 언어

프로그래밍 언어는 GW-BASIC을 사용하였다. 8비트 컴퓨터를 보유하고 있는 경우에는 그 기종에서 사용하는 BASIC언어를 그대로 사용하여도 무난히 실행되도록 하였다. 단지 사용할 수 없는 명령은 time\$ 명령 뿐이다. time\$ 명령은 처리 속도를 조사하기 위하여 사용하였으니 분류 알고리즘을 분석하는데는 사용하지 않아도 좋으리라 생각한다.

3. 프로그램의 구조

프로그램의 구조는 <그림 1>과 같이 주프로그램 부 프로그램으로 나누어 작성하였다.

주프로그램은 분류할 데이터를 생성하고 부프로그램에서 분류한 데이터를 출력하는 부분이고, 부프로그램

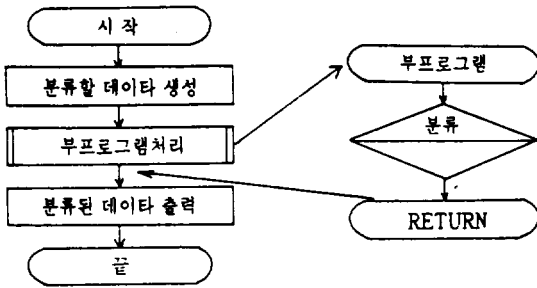


그림 1) 프로그램의 구조

그림은 각종의 내부 분류 프로그램이다.

4. 주프로그램과 부프로그램

(1) 주프로그램

```

100 REM >>>> 주프로그램 <<<<<
110
120 REM ** 분류할 데이터 생성 **
130
140   DEFINT A-Z
150   INPUT "?? NUMBERS OF DATA=" ; N
160   PRINT
170   DIM A(N)
180   PRINT "===> SOURCE DATA LIST"
190   FOR I=1 TO N
200     A(I)=RND(1)*10000
210     PRINT USING"#####";A(I);
220     IF I/10=INT(I/10) THEN PRINT
230   NEXT I
240   PRINT : PRINT
250 REM ** 부프로그램 처리 **
260   GOSUB 300
400
410 REM ** 분류된 데이터 출력 **
420
430   PRINT "===> SORTED DATA LIST"
440   FOR I=1 TO N
450     PRINT USING"#####";A(I);
460     IF I/10=INT(I/10) THEN PRINT
470   NEXT I
480 END
  
```

(2) 부프로그램

① Bubble Sorting(With a Flag)

```

300 REM ** 부프로그램 **
303 '
304 ' <<<<< FLAG SORT >>>>>
  
```

```

305   TIME$=" 00:00:00"
310   C=1
320   S=0
330   FOR I=1 TO N-C
340     IF A(I)<=A(I+1) THEN 350
345     SWAP A(I),A(I+1)
347     S=1
350   NEXT I
360   C=C+1
370   IF S=1 THEN 320
380   PRINT "PROCESS TIME=" ; TIME$
385   PRINT
390   RETURN
  
```

② Selection Sorting

```

300 REM ** 부프로그램 **
303 '
304 REM <<<<< SELECTION SORT >>>>>
305   TIME$=" 00:00:00"
310   FOR I=1 TO N-1
320     B=I
330     FOR J=I+1 TO N
340       IF A(B)>A(J) THEN B=J
350     NEXT J
355     SWAP A(I),A(B)
360   NEXT I
370   PRINT "PROCESS TIME=" ; TIME$
380   PRINT
390   RETURN
  
```

③ Insertion Sorting

```

300 REM ** 부프로그램 **
303 '
304 REM <<<<< INSERTION SORT >>>>>
305   TIME$=" 00:00:00"
310   FOR I=2 TO N
315     X=A(I)
317     B=I-1
320     IF X>A(B) THEN 335
325     A(B+1)=A(B)
327     B=B-1
330     IF B<>0 THEN 320
335     A(B+1)=X
340   NEXT I
350   PRINT "PROCESS TIME=" ; TIME$
360   PRINT
370   RETURN
  
```

④ Shell Sorting

```

300 REM ** 부프로그램 **
303 '
304 ' ◇◇◇ SHELL SORT ◇◇◇
305 TIME$="00:00:00"
310 G=N
315 G=INT(G/2)
320 IF G<1 THEN 380
330 FOR I=G+1 TO N
335 B=I-G
340 IF B<1 THEN 370
350 IF A(B)>A(B+G) THEN 355
353 GOTO 370
355 SWAP A(B),A(B+G)
357 B=B-G
360 GOTO 340
370 NEXT I
375 GOTO 315
380 PRINT "PROCESS TIME=";TIME$
385 PRINT
390 RETURN

```

⑤ Heap Sorting

```

300 REM ** 부프로그램 **
303 '
304 ' ◇◇◇ HEAP SORT ◇◇◇
305 TIME$="00:00:00"
310 B=N
320 FOR L=INT(N/2) TO 1 STEP -1
321 X=A(L)
322 GOSUB 370
325 NEXT L
330 REM == SORT SECTION ==
340 L=1
341 X=A(B) : A(B)=A(1)
342 B=B-1
343 IF B=1 THEN 347
345 GOSUB 370
346 GOTO 341
347 A(1)=X
350 PRINT "PROCESS TIME=";TIME$
355 PRINT
369 RETURN
370 REM == SUBROUTINE ==
371 J=L
372 I=J : J=2*I
375 IF B<J THEN 387
377 IF B=J THEN 381

```

```

378 IF A(J)>=A(J+1) THEN 381
380 J=J+1
381 IF X>A(J) THEN 387
383 A(I)=A(J)
385 GOTO 372
387 A(I)=X
390 RETURN

```

⑥ Quick Sorting

```

300 REM ** 부프로그램 **
303 '
304 ' ◇◇◇ QUICK SORT ◇◇◇
305 TIME$="00:00:00"
307 DIM S(50,2)
310 P=1 : S(P,1)=1 : S(P,2)=N
313 L=S(P,1) : R=S(P,2) : P=P-1
315 GOSUB 350
320 IF R<=L THEN 325
323 P=P+1 : S(P,1)=L : S(P,2)=R
325 R=J
327 IF L<R THEN 315
329 IF P<=0 THEN 313
330 PRINT "PROCESS TIME=";TIME$
335 PRINT
340 RETURN
350 REM == SUBROUTINE ==
360 S=INT((L+R)/2)
363 K=A(S) : I=L : J=R
365 IF K<=A(I) THEN 373
367 I=I+1
370 GOTO 365
373 IF K>=A(J) THEN 380
375 J=J-1
377 GOTO 373
380 IF I>J THEN 390
383 SWAP A(I),A(J)
385 I=I+1 : J=J-1
387 IF I<=J THEN 365
390 RETURN

```

⑦ Radix Sorting

```

300 REM ** 부프로그램 **
303 '
304 ' ◇◇◇ RADIX SORT ◇◇◇
305 TIME$="00:00:00"
310 G=INT(N/2)
315 DIM CN(10),ST(9,G)

```

```

320   FOR P=1 TO 4
321     FOR I=0 TO 3
322       CN(I)=0
323     NEXT I
325     FOR I=1 TO N
326       L1=INT(A(I)/10^(P-1))
327       L=L1-INT(L1/10)*10
328       CN(L)=CN(L)+1
329       ST(L,CN(L))=A(I)
330     NEXT I
335     K=0
340     FOR I=0 TO 9
341       IF CN(I)=0 THEN 347
342       FOR J=1 TO CN(I)
343         K=K+1
344         A(K)=ST(I, J)
345       NEXT J
347     NEXT I
350   NEXT P
360   PRINT "PROCESS TIME=";TIME$
370   PRINT
380   RETURN

```

IV. 실험 및 분석

1. 프로그램의 실행

프로그램은 아래의 순서에 따라 실행시킨다.

① A 드라이브에 MS-DOS디스켓을 꽂고, B 드라이브에는 각종 분류 프로그램이 수록된 디스켓을 꽂은 후 본체의 스위치를 넣으면 MS-DOS가 기동된다. 만일 출력 결과를 프린터로 출력시키려면 이때 프린터의 스위치를 넣는다.

② 날짜와 시간을 입력한 후 「GW-BASIC」을 입력하면 GW-BASIC을 사용할 수 있는 상태가 된다.

③ LOAD "B: SORT.BAS"를 입력하면 주프로그램이 주 기억 장치에 입력된다(주프로그램의 명칭을 SORT.BAS로 했을 경우).

④ MERGE "B: BUBBLE.BAS"를 입력하면 부 프로그램인 BUBBLE SORT 프로그램이 주 프로그램과 병합된다(단, 2개의 프로그램을 하나의 프로그램으로 병합시킬 때에는 프로그램이 디스켓에 ASCII Code로 입력되어 있어야 한다).

⑤ 「RUN」을 입력하면 프로그램이 실행된다.

2. 입출력 형식

입출력 형식은 <그림2>와 같은 형식으로 하였다.

```

RUN
?? NUMBERS OF DATA=? 20

==> SOURCE DATA LIST
1214 6519 8689 7298 7989 737 4903 4545 1072 9505
7039 5319 9712 3209 9561 9345 5349 5644 6712 7026

```

PROCESS TIME=00:00:01

```

==> SORTED DATA LIST
737 1072 1214 3209 4545 4903 5319 5349 5644 6519
6712 7026 7039 7298 7989 8689 9345 9505 9561 9712
Ok

```

<그림 2> 입출력형식

3. 비교 및 분석 결과

위에서 설명한 각종 분류 방법들은 주어진 여건에 따라 여러면으로 영향을 많이 받게 된다. 그러므로 특정한 문제에 어떤 분류 방법이 적절한지 결정하기 위하여 효율에 관한 여러 가지 고려 사항들을 파악해야 한다. 이들 고려 사항 중 가장 중요한 세가지 사항은 첫째 분류 프로그램을 작성하는데 소비하는 시간이며, 둘째 프로그램을 수행하는데 소비되는 시간이고, 셋째 프로그램에 소요되는 기억 공간의 양이다.

이 중에서 가장 중요시되는 것은 분류하는데 소요되는 시간이다. 왜냐하면 대부분의 분류 프로그램에 소요되는 공간은 $O(n^2)$ 보다 $O(n)$ 에 가깝기 때문이고 더 많은 공간이 필요하면 보조 기억 장치를 사용하면 되기 때문이다.

본 연구에서는 각종 분류 프로그램을 실행하여 분류하는데 소요되는 수행 시간을 조사하여 <표1>에 나타내었다(초단위 이하는 무시하였음).

<표 1> 분류 소요 시간

분류방법 n의크기	Bubble Sorting	Selection Sorting	Insertion Sorting	Shell Sorting	Heap Sorting	Quick Sorting	Radix Sorting
10	0	0	0	0	0	0	2
20	2	1	2	1	1	1	3
30	5	2	5	2	2	1	5
40	9	5	9	3	4	2	6
50	14	7	15	4	5	3	7
60	20	11	22	5	6	3	9
70	28	15	30	7	8	4	10
80	36	19	39	8	9	5	12

90	46	25	50	8	11	6	13
100	57	31	71	9	13	6	15
200	227	122	246	23	30	14	29
300	511	273	556	39	49	22	44
400	907	485	984	54	70	31	59
500	1417	757	1538	72	91	37	73
600	2039	1089	2214	91	113	48	87
700	2775	1481	3014	110	135	59	101
800	3624	1933	3937	125	158	68	116
900	4586	2446	4982	148	182	74	130
1000	5661	3019	6149	165	204	80	145

(a) 최악인 경우의 분류 소요 시간(단위 : 초)

70	0	14	0	4	9	3	10
80	0	18	0	5	11	4	12
90	0	23	1	5	13	5	13
100	0	28	1	6	15	6	14
200	1	111	2	14	35	13	29
300	1	249	3	25	57	21	43
400	2	690	4	34	80	30	58
500	2	693	6	43	103	35	72
600	3	993	7	58	128	46	87
700	4	1351	8	68	153	56	101
800	4	1764	9	78	178	65	115
900	5	2232	10	88	204	71	130
1000	5	2754	12	98	230	76	144

(c) 최선인 경우의 분류 소요 시간(단위 : 초)

분류방법 n의 크기	Bubble	Selection	Insertion	Shell	Heap	Quick	Radix
10	0	0	0	0	0	0	2
20	1	1	1	1	1	1	3
30	4	2	3	2	3	2	4
40	7	4	6	3	4	3	6
50	10	7	8	4	6	4	7
60	15	10	11	5	7	5	9
70	21	14	16	7	9	6	10
80	28	18	22	9	10	7	12
90	35	23	26	9	12	8	13
100	44	28	33	12	14	9	14
200	173	111	128	29	32	20	29
300	378	247	271	50	52	35	43
400	673	439	475	74	74	45	57
500	1066	684	767	90	96	59	71
600	1544	984	1121	115	119	72	86
700	2107	1338	1537	130	143	83	100
800	2750	1746	1997	178	167	104	114
900	3469	2208	2508	206	191	111	128
1000	4287	2725	3113	223	216	126	142

(b) 일반적인 경우의 분류 소요시간(단위 : 초)

분류방법 n의 크기	Bubble	Selection	Insertion	Shell	Heap	Quick	Radix
10	0	0	0	0	0	0	2
20	0	1	0	0	2	1	3
30	0	2	0	1	3	1	4
40	0	4	0	2	4	2	6
50	0	7	0	2	6	2	7
60	0	10	0	3	8	3	9

V. 결론

각종 분류 프로그램의 분류 수행 시간은 <표 1>에 나타난 것처럼 분류할 데이터의 형태에 따라 달라질 수 있다.

분류할 데이터가 거의 분류되어 있는 상태라면 <표 1>의 (c)의 분석 결과와 같이

- ① Bubble sorting
- ② Insertion Sorting
- ③ Quick Sorting
- ④ Shell Sorting
- ⑤ Radix Sorting(분류할 데이터의 최대 자릿수가 4자리인 경우)

- ⑥ Heap Sorting
 - ⑦ Selection Sorting
- 의 순으로 처리 시간이 소요되고, 분류할 데이터가 역순으로 나열되어 있을 경우에는 <표 1>의 (a)의 분석 결과와 같이

- ① Quick Sorting
- ② Radix Sorting(최대 4자리일 때)
- ③ Shell Sorting
- ④ Heap Sorting
- ⑤ Selection Sorting
- ⑥ Bubble Sorting
- ⑦ Insertion Sorting

의 순으로 처리 시간이 소요되며, 일반적인 경우에는 <표 1>의 (b)의 분석 결과와 같이

- ① Quick Sorting
- ② Radix Sorting(최대 4자리일 때)
- ③ Heap Sorting
- ④ Shell Sorting
- ⑤ Selection Sorting
- ⑥ Insertion Sorting
- ⑦ Bubble Sorting

의 순으로 처리 시간이 소요되는 것을 알 수 있다.

그리고 데이터의 양이 적을 경우에는 각 분류 프로그램의 수행 시간이 별로 차이가 없으나 데이터의 양이 많을수록 프로그램 간의 차이가 엄청난 사실도 알 수 있다.

위의 분석 결과에서 알 수 있듯이 데이터의 양이나 상태에 따라서 효율성의 높고 낮음이 달라질 수 있다. 따라서 분류의 효율을 극대화시키기 위해서는 분류 기법을 폭넓게 숙지하고, 각기의 장단점을 파악하여 필요할 때 적절한 프로그램을 선택하여 문제를 해결하여야 할 것이다.

참 고 문 헌

1. 구연설·이상호, 「컴퓨터 프로그래밍」, 대은출판사, 1985
2. 강성모, 「고등학교 선산 교육의 현황 분석 및 그 개선 방안」, 경희대학교 교육대학원 석사 학위 논문, 1986
3. 김경태·민용식, 「자료 구조」, 유봉출판사, 1986
4. 이제선, 「국민학교에서의 마이크로 컴퓨터 교육 및 활용에 관한 실태 조사 연구」, 이화여자대학교 교육대학원 석사 학위 논문, 1985
5. 이병욱, 「자료 구조론」, 상조사, 1985
6. 정보분석연구회, 「퍼스널 컴퓨터 이론·실기 문제집」, 도서출판 한국실리콘, 1986.
7. 홍성수, 「COMPUTER BINARY SORTING과 SEARCHING ALGORITHM에 관한 연구」, 광운공과대학 석사 학위 논문, 1982
8. ELLIS HOROWITZ, 「Fundamentals of Computer Algorithms」, COMPUTER SCIENCE PRESS, 1978
9. Jonathan C. Barron, 「Basic Programming Using Structured Modules」, Holsaunders, 1983

ABSTRACT

Comparison & Analysis of Algorithms in BASIC

Seong-mo Kang

Kyungki Commercial High School

Computer in one of the most tremendous achievements of the modern scientific technique. Not only in government, business, research and education but in our daily life, computers are widely utilized to assist in solving various problems.

With increasing frequency, it is recognized that a right understanding of the computer is necessary: naturally, this recognition places a great emphasis on the computer education.

In Korea computer is chosen either as an optional subject or as a kind of group activity in many schools.

It is the purpose of this study to compare and analyze the internal sorting algorithms which are used frequently in data processing, and to present the results of program analysis, which will make it possible to choose the appropriate sorting algorithm for each data processing.

Generally the algorithms are coded in a language appropriate for structured programming, like PASCAL: however, here the algorithms are expressed in BASIC which is widely used with the personal computers so that the students and the teachers may understand them easily.