201511449 컴퓨터과학과 김동현

1. Programming code(source)
2. #include <stdio.h>
3. #include <stdlib.h>
4. #include <time.h>
5. #define MAX 50000
6. #define SWAP(x,y,t) ((t)=(x), (x)=(y), (y)=(t))//swap 함수 미리설정
7. #define parent(x) (x-1)/2
8. int origin[MAX];
9. int list[MAX];
10. int n;
11. int sorted[MAX];
12. clock\_t start, finish, used\_time=0;
13. int COMPARE(int x,int y){
14. int COMPCOUNT=1;
15. if(x<=y||x>=y)
16. COMPCOUNT++;
17. return COMPCOUNT;
18. }
19. //mergesort
20. void Merge(int list[], int left, int mid, int right)
21. {
22. int i,j,k,l;
23. i=left;j=mid+1;k=left;
24. while(i<=mid && j<=right)
25. {
26. if(list[i]<=list[j]){
27. sorted[k++]=list[i++];
28. }
29. else{
30. sorted[k++]=list[j++];
31. }
32. }
33. if(i>mid){
34. for(l=j;l<right;l++)
35. sorted[k++]=list[l];
36. }
37. else{
38. for(l=i;l<=right;l++)
39. sorted[k++]=list[l];
40. }
41. for(l=left;l<=right;l++)
42. list[l]=sorted[l];
43. }
44. int MergeSort(int list[],int left,int right)
45. {
46. int mid;
47. int count=1;
48. if(left<right)
49. {
50. mid=(left+right)/2;
51. MergeSort(list,left,mid);
52. MergeSort(list,mid+1,right);
53. count+=COMPARE(left,right);
54. Merge(list,left,mid,right);
55. }
56. return count;
57. }
58. //quicksort
59. int partition(int list[],int left, int right)
60. {
61. int pivot=list[left],tmp,low=left,high=right+1;
62. do{
63. do{
64. low++;
65. }while(low<=right&&list[low]<pivot);
66. do{
67. high--;
68. }while(high>=left&&list[high]>pivot);
69. if(low<high) {
70. SWAP(list[low],list[high],tmp);
71. }
72. }while(low<high);
73. SWAP(list[left],list[high],tmp);
74. return high;
75. }
76. int QuickSort(int list[],int left,int right)
77. {
78. int count=1;
79. int num=sizeof(list)/sizeof(list[0]);
80. if(left<right)
81. {
82. int q=partition(list,left,right);
83. QuickSort(list,left,q-1);
84. QuickSort(list,q+1,right);
85. count=count+COMPARE(left,right)\*num;
86. }
87. return count;
88. }
89. int InsertionSort(int list[],int n)
90. {
91. int i,j,key,num=1;
92. for(i=1;i<n;i++){
93. key=list[i];
94. for(j=i-1;j>=0&&list[j]>key;j--){
95. list[j+1]=list[j];
96. }
97. num+=COMPARE(list[j],key);
98. list[j+1]=key;
99. }
100. return num;
101. }
102. int heap(int\*data,int num)
103. {
104. int temp;
105. int i=1;
106. int child=i;
107. int root=parent(child);
108. int count=1;
109. for(i=1;i<num;i++){
110. child=i;
111. while(child>0){
112. root=parent(child);
113. if(data[root]<data[child]){
114. SWAP(data[root],data[child],temp);
115. }
116. child=root;
117. count+=COMPARE(data[root],data[child]);
118. }
119. }
120. return count;
121. }
122. void CopyArr()
123. {
124. int i;
125. for(i=0;i<n;i++)
126. list[i]=origin[i];
127. }
128. void CalcTime()
129. {
130. used\_time=finish-start;
131. printf("소요 시간:%f sec\n",(float)used\_time/CLOCKS\_PER\_SEC);
132. }
133. int main()
134. {
135. int i;
136. n=1024;
137. //n=32;
138. for(i=0;i<n;i++){
139. origin[i]=rand();
140. //origin[i]=32-i;
141. }
142. for(i=0;i<n;i++)
143. printf("%d\t",origin[i]);
144. printf("데이터의 갯수:%d\n",n);
145. CopyArr();
146. start=clock();
147. int counter=InsertionSort(list,n);
148. finish=clock();
149. printf("비교횟수:%d\n",counter);
150. printf("삽입정렬시간~~~\n");
151. CalcTime();
152. CopyArr();
153. start=clock();
154. counter=heap(list,n);
155. finish=clock();
156. printf("비교횟수:%d\n",counter);
157. printf("힙정렬시간~~~\n");
158. CalcTime();
159. CopyArr();
160. start=clock();
161. counter=MergeSort(list,0,n);
162. finish=clock();
163. printf("비교횟수:%d\n",counter);
164. printf("합병정렬시간~~~\n");
165. CalcTime();
166. CopyArr();
167. start=clock();
168. counter=QuickSort(list,0,n);
169. finish=clock();
170. printf("비교횟수:%d\n",counter);
171. printf("퀵정렬시간~~~\n");
172. CalcTime();
173. }

2. input file

데이터의 갯수:1024

41 18467 6334 26500 19169 15724 11478 29358 26962 24464 5705 28145 23281 16827 9961

491 2995 11942 4827 5436 32391 14604 3902 153 292 12382 17421 18716 19718 19895

5447 21726 14771 11538 1869 19912 25667 26299 17035 9894 28703 23811 31322 30333 17673

4664 15141 7711 28253 6868 25547 27644 32662 32757 20037 12859 8723 9741 27529 778

12316 3035 22190 1842 288 30106 9040 8942 19264 22648 27446 23805 15890 6729 24370

15350 15006 31101 24393 3548 19629 12623 24084 19954 18756 11840 4966 7376 13931 26308

16944 32439 24626 11323 5537 21538 16118 2082 22929 16541 4833 31115 4639 29658 22704

9930 13977 2306 31673 22386 5021 28745 26924 19072 6270 5829 26777 15573 5097 16512

23986 13290 9161 18636 22355 24767 23655 15574 4031 12052 27350 1150 16941 21724 13966

3430 31107 30191 18007 11337 15457 12287 27753 10383 14945 8909 32209 9758 24221 18588

6422 24946 27506 13030 16413 29168 900 32591 18762 1655 17410 6359 27624 20537 21548

6483 27595 4041 3602 24350 10291 30836 9374 11020 4596 24021 27348 23199 19668 24484

8281 4734 53 1999 26418 27938 6900 3788 18127 467 3728 14893 24648 22483 17807

2421 14310 6617 22813 9514 14309 7616 18935 17451 20600 5249 16519 31556 22798 30303

6224 11008 5844 32609 14989 32702 3195 20485 3093 14343 30523 1587 29314 9503 7448

25200 13458 6618 20580 19796 14798 15281 19589 20798 28009 27157 20472 23622 18538 12292

6038 24179 18190 29657 7958 6191 19815 22888 19156 11511 16202 2634 24272 20055 20328

22646 26362 4886 18875 28433 29869 20142 23844 1416 21881 31998 10322 18651 10021 5699

3557 28476 27892 24389 5075 10712 2600 2510 21003 26869 17861 14688 13401 9789 15255

16423 5002 10585 24182 10285 27088 31426 28617 23757 9832 30932 4169 2154 25721 17189

19976 31329 2368 28692 21425 10555 3434 16549 7441 9512 30145 18060 21718 3753 16139

12423 16279 25996 16687 12529 22549 17437 19866 12949 193 23195 3297 20416 28286 16105

24488 16282 12455 25734 18114 11701 31316 20671 5786 12263 4313 24355 31185 20053 912

10808 1832 20945 4313 27756 28321 19558 23646 27982 481 4144 23196 20222 7129 2161

5535 20450 11173 10466 12044 21659 26292 26439 17253 20024 26154 29510 4745 20649 13186

8313 4474 28022 2168 14018 18787 9905 17958 7391 10202 3625 26477 4414 9314 25824

29334 25874 24372 20159 11833 28070 7487 28297 7518 8177 17773 32270 1763 2668 17192

13985 3102 8480 29213 7627 4802 4099 30527 2625 1543 1924 11023 29972 13061 14181

31003 27432 17505 27593 22725 13031 8492 142 17222 31286 13064 7900 19187 8360 22413

30974 14270 29170 235 30833 19711 25760 18896 4667 7285 12550 140 13694 2695 21624

28019 2125 26576 21694 22658 26302 17371 22466 4678 22593 23851 25484 1018 28464 21119

23152 2800 18087 31060 1926 9010 4757 32170 20315 9576 30227 12043 22758 7164 5109

7882 17086 29565 3487 29577 14474 2625 25627 5629 31928 25423 28520 6902 14962 123

24596 3737 13261 10195 32525 1264 8260 6202 8116 5030 20326 29011 30771 6411 25547

21153 21520 29790 14924 30188 21763 4940 20851 18662 13829 30900 17713 18958 17578 8365

13007 11477 1200 26058 6439 2303 12760 19357 2324 6477 5108 21113 14887 19801 22850

14460 22428 12993 27384 19405 6540 31111 28704 12835 32356 6072 29350 18823 14485 20556

23216 1626 9357 8526 13357 29337 23271 23869 29361 12896 13022 29617 10112 12717 18696

11585 24041 24423 24129 24229 4565 6559 8932 22296 29855 12053 16962 3584 29734 6654

16972 21457 14369 22532 2963 2607 2483 911 11635 10067 22848 4675 12938 2223 22142

23754 6511 22741 20175 21459 17825 3221 17870 1626 31934 15205 31783 23850 17398 22279

22701 12193 12734 1637 26534 5556 1993 10176 25705 6962 10548 15881 300 14413 16641

19855 24855 13142 11462 27611 30877 20424 32678 1752 18443 28296 12673 10040 9313 875

20072 12818 610 1017 14932 28112 30695 13169 23831 20040 26488 28685 19090 19497 2589

25990 15145 19353 19314 18651 26740 22044 11258 335 8759 11192 7605 25264 12181 28503

3829 23775 20608 29292 5997 17549 29556 25561 31627 6467 29541 26129 31240 27813 29174

20601 6077 20215 8683 8213 23992 25824 5601 23392 15759 2670 26428 28027 4084 10075

18786 15498 24970 6287 23847 32604 503 21221 22663 5706 2363 9010 22171 27489 18240

12164 25542 7619 20913 7591 6704 31818 9232 750 25205 4975 1539 303 11422 21098

11247 13584 13648 2971 17864 22913 11075 21545 28712 17546 18678 1769 15262 8519 13985

28289 15944 2865 18540 23245 25508 28318 27870 9601 28323 21132 24472 27152 25087 28570

29763 29901 17103 14423 3527 11600 26969 14015 5565 28 21543 25347 2088 2943 12637

22409 26463 5049 4681 1588 11342 608 32060 21221 1758 29954 20888 14146 690 7949

12843 21430 25620 748 27067 4536 20783 18035 32226 15185 7038 9853 25629 11224 15748

19923 3359 32257 24766 4944 14955 23318 32726 25411 21025 20355 31001 22549 9496 18584

9515 17964 23342 8075 17913 16142 31196 21948 25072 20426 14606 26173 24429 32404 6705

20626 29812 19375 30093 16565 16036 14736 29141 30814 5994 8256 6652 23936 30838 20482

1355 21015 1131 18230 17841 14625 2011 32637 4186 19690 1650 5662 21634 10893 10353

21416 13452 14008 7262 22233 5454 16303 16634 26303 14256 148 11124 12317 4213 27109

24028 29200 21080 21318 16858 24050 24155 31361 15264 11903 3676 29643 26909 14902 3561

28489 24948 1282 13653 30674 2220 5402 6923 3831 19369 3878 20259 19008 22619 23971

30003 21945 9781 26504 12392 32685 25313 6698 5589 12722 5938 19037 6410 31461 6234

12508 9961 3959 6493 1515 25269 24937 28869 58 14700 13971 26264 15117 16215 24555

7815 18330 3039 30212 29288 28082 1954 16085 20710 24484 24774 8380 29815 25951 6541

18115 1679 17110 25898 23073 788 23977 18132 29956 28689 26113 10008 12941 15790 1723

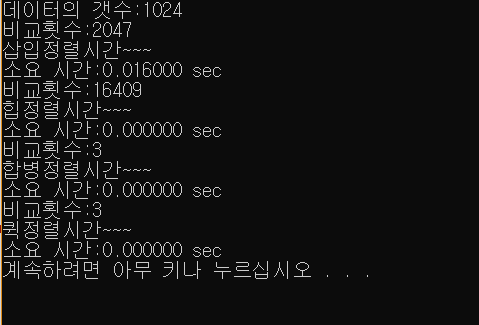
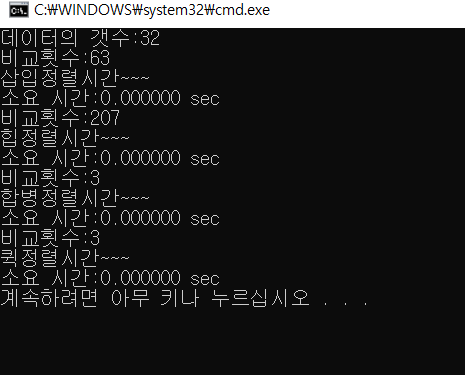
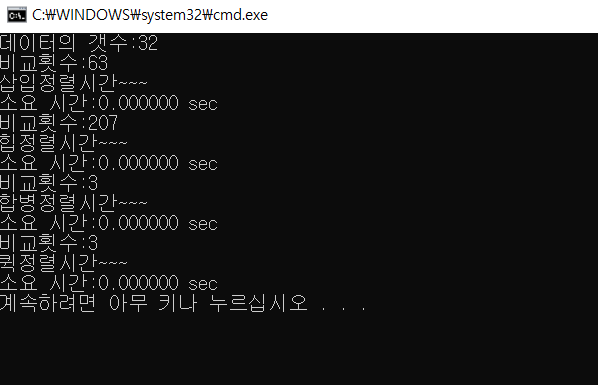
21363 28 25184 24778 7200 5071 1885 21974 1071 11333 22867 26153 14295 32168 20825

9676 15629 28650 2598 3309 4693 4686 30080 10116 12249 26667 1528 26679 7864 29421

8405 8826 6816 7516 27726 28666 29087 27681 19964 1340 5686 6021 11662 14721 6064

29309 20415 17902 29873

3.output file



4.documentation

총 9개의 함수로 구현 하였는데

1. Insertion sort 한 개
2. Heap sort 한 개
3. Merge, MergeSort 함수
4. Partition, QuickSort 함수
5. 시간복잡도 측정함수

시간복잡도는 time.h의 clock()함수를 이용하였다.

1. 원본배열 복사함수
2. 비교횟수 체크 함수

메인함수에서 인풋을 만들어 계속 넣어 봄으로서 시간복잡도와 비교횟수를 체크 하였다.

5. Lessons learned from programming assignment

데이터 개수가 작을 때(n=32)는 insertion sort가 빨랐으나 데이터 개수가 많아질수록 힙정렬, 합병정렬, 퀵정렬이 빠르다는 것을 몸소 느꼈다. 그리고 큰 양의 데이터를 넣어 봄으로서 덩달아 나의 컴퓨터 계산속도도 가늠해볼 수 가 있었다. 다시 한번 수업시간에 배운 알고리즘들을 복습하는 유익한 시간이 되었다.