

COMP319 Algorithms 1, Spring 2021

Homework Programming Assignment 3 (HW3)

알고리즘1, 2021년 봄학기, 프로그래밍 숙제 3

Instructor: Gil-Jin Jang Email: gjang@knu.ac.kr
School of Electronics Engineering, Kyungpook National University
장길진, 경북대학교 전자공학부

프로그래밍 숙제 3번의 목적은 다음과 같다.

1. Quick sort, merge sort를 이용한 문자열 정렬 알고리즘 구현
2. 연속된 메모리 공간에 순차적으로 저장된 문자열 정렬 알고리즘 구현
3. 문자열의 길이가 고정되어 있을 경우와 가변 길이의 정렬 알고리즘 구현

공통 요구사항:

1. `int main(int argc, char *argv[]) { ... }` 로 main 함수가 작성되어야 하며, `./hw3-x.exe` 로 실행된다. (어차피 다시 컴파일할 것이기 때문에 `.c` 파일만 제대로 제출되면 문제없다)
2. 처음의 주석문에는 본인의 정보가 기재되어야 함.
3. merge/quick sort는 작성되어 있는 코드가 굉장히 많으며, 강의자료에도 있다. 다른 학생의 코드와 거의 동일하면 COPY 판정을 받을 수 있기 때문에, 본인이 기존의 코드를 참조했으면 웹주소, 책이름, 강의자료 차수와 쪽수 등을 주석문으로 넣는다. 만약 다른 학생의 코드와 유사한데 참조정보가 없으면 COPY 판정을 받게 된다.
4. 한글문자는 코드에 넣지 않는다. 컴파일 오류를 발생시킬 수 있으며, 실행점수를 전혀 못 받을 수도 있음. 이름도 영어로 적는 것을 권장한다.

1 Homework 3-1

입력파일예: 다음과 같이 무작위로 생성된 길이가 고정되어 있는 단어들이 '<' 와 '>' 의 사이에 저장되어 있다. 입력파일은 문제를 간단히 하기 위해서 줄바꿈, 빈칸, 탭 문자는 없고 영소문자와 단어구분을 위한 '<', '>'로만 되어 있다. 정확한 입력 파일의 형식을 이해하기 위해서 주어진 예제 입력 파일을 열어보면 다음과 같다.

FILE: fix8input_9.txt (fixed length 8, 9 words)

<hlzwhhls><xzpsqrq><oahdumjx><ztnvojms><gfdlpfbg><ypghwygb><ncpbvmxb><nkpjrgbk><giadfjzi>

FILE: fix8input_99.txt (fixed length 8, 99 words)

<nopqilox><uvhtwlaz><pltupaxn><wyxzhws><hvogijrf><ddygnjxl><qbugucwl><zmsocnra><uzwzqomn><ahjkfjfg><vyufuhs><xzmiwwkt><azsyaygo><prznpiqv><xjfwzgs><bunmxlfl><aezcjpcu><mtkmdidw><qixanqj><miafwctb><btauvqsm><fxohdeup><atyjupsq><kazzudov><ibkztfu><bsthssc><dfrikeoh><psicvlpf><qjhinhfk><nybxkpw><spnlrqjx><ngyswygk><wkeuemnw><htjqsik><saerormi><tctxnnnc><udvochjq><fgnqzrol><ssrehimm><fwfqzhuq><jpfrxjfv><

```

ojpcdmyb><oqcpxac><hevhpk><xwuatiux><iehftbon><zhxykpxp><qeliemca><phljwxid><
aurjcdzk><ffkppoor><zpjoubr><kyxghkxa><htbmrmt><kwoefesm><brzuprzh><jilbsgbr><
qvxhndpf><vwwliqz><wbqsnfcf><tlditrzb><bxuhnyau><yhpaoyiq><nrtiqrkp><rwvosejb><
oxfnxaw><dmwookrp><yjbwzwxg><saqenqje><kzhyxxnx><unhcwusu><ohudtsoc><sdzjdjpg><
yqzbepnr><wjcoycqb><kuozgysu><ixwnfajg><scxihxkp><lwekhmie><xjtwyiam><ttgfjgca><
jzfmuy><abxtiui><luvqqkbh><nbrfpmav><fizxdzhg><zgjbdrmx><cooislr><wigvyxnn><
phceryal><hdjrokhm><neqbld><fbzplkxe><rpihjypj><zytpkgnp><mcwqcedv><cgooesex><
pecvsjxv><qynmceem>

```

FILE: fix16input_9.txt (fixed length 16, 9 words)

```

<lyvgitifnrsljryn><lkwydnagaglxvbx><gxrzpywxpxkyghdv><qhxngnabsjzbxulp><lrftgxcugpsgytm><
uqndttsumaxexrk><vyqkwfnisyaopqob><yqvpogwsudazxrve><xgqsczwiplvmrkyb>

```

할것들: 주어진 텍스트 파일을 한꺼번에 메모리로 올리고(char array), 메모리의 단어들을 **merge sort**와 **quick sort**를 이용하여 오름차순으로 정렬하고, 이를 다시 텍스트 파일로 저장하는 프로그램을 작성한다.

실행예: 채점에 사용되는 command line argument 는 정렬방법/입력파일명/출력파일명이다.

```

$ ./hw3-1.exe 1 fix8input_9.txt sort1_fix8input_9.txt
0.00003 seconds
185 bytes ( 2.011 x 92 )

$ ./hw3-1.exe 2 fix8input_9.txt sort2_fix8input_9.txt
0.00002 seconds
103 bytes ( 1.120 x 92 )

$ ./hw3-1.exe 1 fix8input_99.txt sort1_fix8input_99.txt
0.00006 seconds
1985 bytes ( 2.001 x 992 )

$ ./hw3-1.exe 2 fix8input_99.txt sort2_fix8input_99.txt
0.00009 seconds
1003 bytes ( 1.011 x 992 )

$ ./hw3-1.exe 1 fix8input_999.txt sort1_fix8input_999.txt
0.00077 seconds
19985 bytes ( 2.000 x 9992 )

$ ./hw3-1.exe 2 fix8input_999.txt sort2_fix8input_999.txt
0.00068 seconds
10003 bytes ( 1.001 x 9992 )

$ ./hw3-1.exe 1 fix16input_99.txt sort1_fix16input_99.txt
0.00007 seconds
3567 bytes ( 2.001 x 1783 )

$ ./hw3-1.exe 2 fix16input_99.txt sort2_fix16input_99.txt
0.00007 seconds
1802 bytes ( 1.011 x 1783 )

$ ./hw3-1.exe 1 fix16input_999.txt sort1_fix16input_999.txt
0.00102 seconds
35967 bytes ( 2.000 x 17983 )

$ ./hw3-1.exe 2 fix16input_999.txt sort2_fix16input_999.txt
0.00077 seconds
18002 bytes ( 1.001 x 17983 )

```

설명:

1. 방법(argv[1]): 1 - merge sort, 2 - quick sort
2. 입력파일(argv[2]): 설명 참고
3. 출력파일(argv[3]): 입력파일과 같은 형식으로 저장되며, 정렬되어 있어야 한다.
4. 고정단어길이: 입력파일에서 유추한다. 첫 단어에서 '<'와 '>' 사이의 문자 개수를 세면 된다.
5. 화면출력줄1: 마지막 예제의 경우 **0.00077 seconds** 측정된 소요시간
6. 화면출력줄2: 마지막 예제의 경우 **18002 bytes (1.001 x 17983)** 측정된 사용 메모리 고정길이 16이므로 '<>'을 포함하면 18 bytes/word, 단어는 999개이므로 $999 \times 18 = 17982$ bytes가 입력메모리의 크기이며, 총 사용된 메모리는 입력 메모리 17982 bytes 포함하여 18002 bytes이므로 비례크기는 **1.001**

출력파일에: 입력파일과 같은 형식이지만 알파벳 순서로 정렬되어 있음

FILE: sort1_fix8input_9.txt (sort2_fix8input_9.txt 와 동일, 줄바꿈 없음)
<gfdlpfbg><giadfjzi><hlzwhhls><ncpbvmxb><nkpjrgbk><oahdumjx><xzxpqrq><ypghwygb><ztnvojs>

FILE: sort1_fix8input_99.txt (sort2_fix8input_99.txt 와 동일, 줄바꿈 없음)
<abxtiui><aezcpjcu><ahjkfjfg><atyjupsq><aurjcdzk><azsyaygo><brzuprzh><bsthssc><btavqvm><bunmxlfl><bxuhnyau><cgooesex><cooislr><ddygnjxl><dfrikeoh><dmwookrp><fbzplkxe><ffkppoor><fgnqzrol><fizxdzhg><fwfqzhuq><fxohdeup><hdjrokhm><hevhokpr><htbmrmt><htjqsii><hvogijrf><ibkztfuo><iehftbon><ixwnfajg><jilbsgbr><jprfxjfv><jzfzmuym><kazzudov><kuozgysu><kwoefesm><kxyghkxa><kzhyxxnx><luvqqkbh><lwekhmie><mcwqcedv><miafwctb><mtkmdidw><nbrfpnav><neqbldsd><ngyswygk><nopqilox><nrtiqrkp><nybxkpw><ohudtsoc><ojpcdmyb><oqcppxac><oxfnxwaw><pecvsjxv><phceryal><phljwxid><pltupaxn><prznpiqv><psicvlpf><qbugucwl><qeliemca><qixanqj><qjhinfhk><qvxhndpf><qynmceem><rpjhjypj><rwvosejb><saerormi><saqenqje><scxihxkp><sdzdzjg><spnlrqjx><ssrehimm><tctxnnnc><tlditrzb><tgtfjgca><udvochjq><unhcwswu><uvhtw laz><uzwzqomn><vwlwizq><vyufuhs><wbqsnfcf><wigvyxnn><wjcoyqcb><wkeuemnw><wyxztwhs><xjfwzqgs><xjtwyiam><xwuatiux><xzmiwwkt><yhpaoyiq><yjwbzwxg><yqzbepnr><zgjbdrmx><zhykpxp><zmsocnra><zpjoubrr><zytpkgnp>

FILE: sort1_fix16input_9.txt (sort2_fix16input_9.txt 와 동일)
<gxrzpywxpxkyghdv><lkwydnagaglxvbx><lrtfgxjcugpsgytm><lyvgitifnrsljryn><qhxngnabsjzbxulp><uqndttsumaxexrk><vyqkwfnisyaopqob><xgqsczwlplvmrkyb><yqvpogwsudazzrve>

1. 파일 입출력이 구현되어 있으며, read_bytes(), write_bytes() 이외의 파일 입출력은 사용할 수 없음(감점요인)
read_bytes() 입력 파일이름이 인자로 주어지며 메모리를 malloc.c 함수를 이용하여 파일크기만큼 할당하고 fread 함수로 통째로 읽어들이. 파일의 크기는 포인터 변수를 통해 전달함
write_bytes() 출력 파일이름이 인자로 주어지며 처리가 끝난 메모리를 fwrite 함수를 이용하여 통째로 저장함
NOT ALLOWED 이 두 함수만을 이용하여 파일 입출력이 허용됨
2. "FILL" 로 표시가 되어 있는 영역에 코드 구현, 추가 함수 작성은 허용됨(해야함)
3. 정확한 메모리 사용량을 측정하기 위해서는 배열을 사용할 경우에는 **malloc.c()** 함수를 사용한다. 불필요한 감점을 받지 않기 위하여 **calloc**, **realloc** 등 다른 메모리 할당 함수, 혹은 **strdup** 등 메모리 할당함수는 사용하지 않는다. 대신 필요할 경우 template에 정의되어 있는 **strdup_c**를 사용한다. 배열이 아닌 일반 변수는 사용량에서 무시한다.
4. **memcpy**, **memccpy**, **memmove**, **wmemmove** 등의 함수는 사용하지 않는다.(cheating으로 간주되어 시간 점수 0점이 부여될 수 있음) 이 함수들은 시스템 HW call 등을 이용하여 매우 효율적으로 구현되어 있으며 UNIT OPERATION TIME 가정에 위해되기 때문에 제대로 평가를 할 수 없다.
5. **strtok** 함수는 사용해도 되지만 안 써도 충분히 구현가능하다(사용하지 않는 것을 추천함)

제출물: hw3-1.c, 오류 없이 컴파일 가능, 위의 실행예에 따라 실행가능해야 하며, 위의 예제의 형식에 맞는 입/출력 파일을 생성해야 한다.

채점기준 및 평가방법

10점 제출 기본점수

10점 학번/이름 주석문에 정확히 입력

40점 컴파일 오류 없고 실행이 정확한지 평가

- 예제로 주어진 입력과 다른 입력에 대하여 정확한지 평가. 매우 큰 파일로도 평가함. 각 예제 별로 정답과 완전히 동일해야 한다(한 글자도 틀리면 그 입력에 대하여 0점)
- 컴파일 오류시 실행불가로 0점이 주어짐
- 답이 맞더라도 알고리즘을 제대로 구현하지 않으면 해당과제를 하지 않은 것으로 판단하여 실행점수 0점

10점 실행시간 점수

- 10/5/0 3단계
- 10: 다른 학생들에 비해 월등하게 빠름
- 5: 거의 비슷함
- 0: 너무 오래 걸림
- 주관적이지만 큰 문제없이 알고리즘을 구현하였을 경우, 즉 대부분의 학생들이 5점을 받도록 분포를 보고 결정하겠습니다.

10점 메모리 사용량 점수,

$$10 \times \frac{N}{M}$$

N : 입력 파일의 bytes 수, M : 사용한 메모리량. 즉, 추가메모리를 전혀 사용하지 않아야 10점 받을 수 있음. 예를 들어 '**3799963 bytes (2.235 x 1699983)**' 의 경우

$$10 \times \frac{1699983}{3799963} = 10/2.235 = 4.474$$

출력된 메모리 사용량은 TA가 코드를 보고 누락된 것이 없는지 검증한다. 필요시 코드를 수정하거나, 수작업으로 값을 업데이트할 수 있으며 작성한 코드가 너무 복잡하여 메모리 사용량을 알 수 없을 경우 '판독불가'로 평가하고 점수를 부여하지 않는다.

20점 코드에 대한 평가(조건들 만족. 영문만을 사용/소수점/입출력 등)

COPY 적발시 전체 0점. COPIED/BEING COPIED 모두 해당됨.

2 Homework 3-2

입력파일예: 3-1의 입력과 유사하지만 다음과 같이 **가변 길이**의 무작위로 생성된 단어들이 '.'를 이용하여 구분되어 저장되어 있다.

FILE: varinput_9.txt (variable length, 9 words)

<bivlcg><fit><lnjbhet><cg><ibqiyby><ewobqu><tbpiibhp><bbilqyk><febhw>

FILE: varinput_99.txt (fixed length 8, 99 words)

<kkrgu><gqntdrtdttofud><spqlhvsjhokabmpd><zi><cgchkrytea><xmwwryqmkhseq><kkx><phapbez><blaedmljxxgmmirj><vwgmswx><mtzdpjpkhcueew><s><z><lnn><vjgrjmlghiy><jouyapi><qqohbpmw><bzddhsovoaa><rsljzmz><ifixisdjzbvwpvv><s><mmatdosqxu><brh><izyymmmax><fclhfiqtzbbdoi><xyiviepuazpxe><fzuhlsvngx><sntmavhlis><bcmdepmfcohom><rcsnbz><nil><kmwmifezyl><dbi><jc><duig><obr><qbm><sbpokycttouiudp><hreaqpjzvekrsdqg><sknzduwjdrlimx><tvxfjdhienw><sqpzobxnhgrtvtbvk><obqjtgfhk><kpa><zclwtwineqokroqn><wwlqtyvmbou><sqztlztmkhenlpo><en><zacoxezxxf><bthcwmsdiskjbb><jijouqcla><pwpa><zzzilp><fqhpgzegtiwstqei><tvshdkefwysbglop><zftztzqolzfjh><fdrfjvvenqs><wbhplfdpcnlqadta><gfithoeliirxv><rxthnisaiaeqqfpw><yc><qrtdp><iepfucqxmsu><fmxnpbejmlgf><mafmgwb><gggxp><scgodxelepnaazw><qli><xlm><apwyskticz><wqgznejeig><piyyrceqlzye><jsbj><uslyksobc><mopu><dnb><ybz><vbzarcaefxoce><tjcvfldtxbfwzsq><dd><wk><vodege><thbi><rgfpbfgiadvz><qzwejjzdhqvhujvqo><puwiamluutdmhhe><noilncgx><fvdhrjj><v><sfhhng><rxgddlncamxiufk><ziproncw><mswjxptfihjcsu><ntsawd><medgcqtpmdz><vqhr><rglvkflm><ech><jn>

할것들: 가변 단어들을 **merge sort**와 **quick sort**를 이용하여 오름차순으로 정렬하고, 이를 다시 텍스트 파일로 저장하는 프로그램을 작성한다.

실행예: 채점에 사용되는 command line argument 는 정렬방법/입력파일/출력파일이다.

```
$ ./hw3-2.exe 1 varinput_9.txt sort1_varinput_9.txt
0.00001 seconds
215 bytes ( 3.028 x 71 )
```

```
$ ./hw3-2.exe 2 varinput_9.txt sort2_varinput_9.txt
0.00001 seconds
179 bytes ( 2.521 x 71 )
```

```
$ ./hw3-2.exe 1 varinput_99.txt sort1_varinput_99.txt
0.00007 seconds
2911 bytes ( 2.749 x 1059 )
```

```
$ ./hw3-2.exe 2 varinput_99.txt sort2_varinput_99.txt
0.00011 seconds
2515 bytes ( 2.375 x 1059 )
```

```
$ ./hw3-2.exe 1 varinput_999.txt sort1_varinput_999.txt
0.00071 seconds
21035 bytes ( 3.226 x 6521 )
```

```
$ ./hw3-2.exe 2 varinput_999.txt sort2_varinput_999.txt
0.00091 seconds
17039 bytes ( 2.613 x 6521 )
```

설명:

1. 방법(argv[1]): 1 - merge sort, 2 - quick sort
2. 입력파일(argv[2]): 설명 참고
3. 출력파일(argv[3]): 입력파일과 같은 형식으로 저장되며, 정렬되어 있어야 한다.
4. 화면출력줄1: 마지막 예제의 경우 **0.00091 seconds** 측정된 소요시간
5. 화면출력줄2: 마지막 예제의 경우 **17039 bytes (2.613 x 6521)** 측정된 사용 메모리 가변길이
이므로 단어의 개수에 비례하지 않으며 6521 bytes 가 입력메모리의 크기이고, 총 사용된 메모리는
17039 bytes 이므로 비례크기는 2.613 (입력 메모리 6521 bytes 포함)

출력파일예: 입력파일과 같은 형식이지만 알파벳 순서로 정렬되어 있음

FILE: sort1_varinput_9.txt (sort2_varinput_9.txt 와 동일, 줄바꿈 없음)
<bbilqyk><bivlbg><cgm><ewobqu><febjw><fit><ibqiyby><lnjbhet><tbpibhp>

FILE: sort1_varinput_99.txt (sort2_varinput_99.txt 와 동일, 줄바꿈 없음)
<apwyskticz><bcmdepfcmfcohom><blaedmljxxgmmirj><brh><btchcwmsdiskjbb><bzddhsovooa><cgchkrytea><dbi><dd><dnb><duig><ech><en><fclhfiqtzbbdoi><fdrfjvvenqs><fmnxpbejmlgf><fqhpgzegtiwstqei><fvdhrjj><fzuhlsvgw><gfithoeliirxv><ggqxp><gqntdrtdltdtofud><hreaqpjzvekrdsdgg><iepfucqmxmsu><ifxisdjzbbvwpv><izyymmmax><jc><jijouqcla><jn><jouyapi><jsbj><kkrgu><kxx><kmwmifezyl><kpa><lnn><mafmwgb><medgcqtpmdz><mmatdosqxu><mopu><mswjxptfihjcsu><mtzdpjpkhcueew><nil><noilncgx><ntsawd><obqgjtghfk><obr><phapbez><piyyrceqlzye><puwiamluutdmhhe><pupa><qbm><qli><qqohbpmw><qrtdp><qzwejjzdhqvhujuvqo><rcsnbz><rgfpbfgiadzv><rglvkflm><rsljmz><rxgddlncamxiufk><rxthnisaiaeqqfpw><s><s><sbpokycttouiudp><scgodxelepnaazw><sfhhng><sknzduwjdrlimx><sntmavhlis><spqlhvsjhokabmpd><sqpzobxnhgrtvbwk><sqztlztmkhenlpo><thbi><tjcvfldtxbzfwszsq><tvshdkefwysbglop><tvxfejdhienw><uslyksobc><v><vzbzarcaefxoce><vjgrjmlghiy><vodege><vqhr><vwgmswx><wbhplfdpcnlqadta><wk><wqgznejeig><wwlqtyvmbou><xlm><xmwryqmkhseq><xyiviepueazpxe><ybz><yc><z><zacoxezxxf><zclwtwineqokroqn><zftztzqolzafjh><zi><ziproncw><zzzilp>

1. Homework3-1과 같은 template code 사용
2. 요구조건들은 3-1과 같음

제출물: hw3-2.c, 오류 없이 컴파일 가능, 위의 실행예에 따라 실행가능해야 하며, 위의 예제의 형식에 맞는 입/출력 파일을 생성해야 한다.

채점기준 및 평가방법

10점 제출 기본점수

10점 학번/이름 주석문에 정확히 입력

40점 컴파일 오류 없고 실행이 정확한지 평가

- 예제로 주어진 입력과 다른 10가지 이상의 입력에 대하여 정확한지 평가. 매우 큰 파일로도 평가함. 각 예제별로 정답과 완전히 동일해야 한다(한 글자도 틀리면 그 입력에 대하여 0점)
- 컴파일 오류시 실행불가로 0점이 주어짐
- 답이 맞더라도 알고리즘을 제대로 구현하지 않으면 해당과제를 하지 않은 것으로 판단하여 실행점수 0점

10점 실행시간 점수

- 10/5/0 3단계
- 10: 다른 학생들에 비해 월등하게 빠름
- 5: 거의 비슷함
- 0: 너무 오래 걸림
- 주관적이지만 큰 문제없이 알고리즘을 구현하였을 경우, 즉 대부분의 학생들이 5점을 받도록 분포를 보고 결정하겠습니다.

10점 메모리 사용량 점수,

$$10 \times \frac{N}{M}$$

N : 입력 파일의 bytes 수, M : 사용한 메모리양. 즉, 추가메모리를 전혀 사용하지 않아야 10점 받을 수 있음. 예를 들어 '2301733 bytes (2.421 x 950868)' 의 경우

$$10 \times \frac{950868}{2301733} = 10/2.421 = 4.13$$

출력된 메모리 사용량은 TA가 코드를 보고 누락된 것이 없는지 검증한다. 필요시 코드를 수정하거나, 수작업으로 값을 업데이트할 수 있으며 작성한 코드가 너무 복잡하여 메모리 사용량을 알 수 없을 경우 '판독불가'로 평가하고 점수를 부여하지 않는다.

20점 코드에 대한 평가(조건들 만족. 영문만을 사용/소수점/입출력 등)

COPY 적발시 전체 0점. COPIED/BEING COPIED 모두 해당됨.

3 제출형식 및 방법

template 코드 및 요구조건: template-hw3.c, hw3-input-sorted.zip

제출할 코드 hw3-1.c, hw3-2.c 만을 제출해야 함. 입출력 파일은 제출하지 않는다. 필요없는 파일 제출시 10점까지 감점 있음

제출방법 위의 2개의 파일을 묶어서 hw3.zip 을 만들고, lms.knu.ac.kr 에 업로드한다. LMS는 제출된 파일의 이름을 복잡하게 바꾸기 때문에 개별파일을 제출하면 채점이 매우 어렵다.