

Algorithm HW #1

컴퓨터소프트웨어학부 2018008231 김효은

컴파일 환경: g++ (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0

1-1. Heap sort

※ 파일 이름: 1-1HeapSort.cpp, 1-1HeapSort.out

1) 헤더 파일 목록

- <iostream>

2) Functions

function	설명
Parent(int x): int	index가 x인 node의 부모 index를 찾아서 return
LChild(int x): int	index가 x인 node의 왼쪽 자식 index를 찾아서 return
RChild(int x): int	index가 x인 node의 오른쪽 자식 index를 찾아서 return
swap(int* a, int* b): void	a와 b의 값을 서로 바꾼다.
Max_Heapify(int nodenum, int size, int* array): void	index가 nodenum인 node가 Heap의 규칙에 알맞도록 array를 정렬한다. 이 때 array의 크기는 size로 받는다.

3) main 함수 동작 과정

- ① stdin으로 n개의 자연수를 입력 받아 array에 저장한다.
- ② array의 가장 마지막 원소의 부모부터 루트 노드까지 Heap의 규칙에 알맞도록 정렬한다.
- ③ Heap의 가장 첫 노드를 꺼내어 출력하고 Heap을 다시 정렬한다.
- ④ 모든 원소를 출력할 때까지 과정 ③을 반복한다.

4) 실행 화면 캡처

- 주어진 예시

```
sp2019@sp2019-VirtualBox:~/algorithm$ g++ -o 1-1HeapSort.out 1-1HeapSort.cpp
sp2019@sp2019-VirtualBox:~/algorithm$ ./1-1HeapSort.out
Input numbers to sort:
9 45 871 23 13 13 88 46 12 51 99 -1

Result:
871 99 88 51 46 45 23 13 13 12 9
```

- n=99일 때

```
sp2019@sp2019-VirtualBox:~/algorithm$ ./1-1HeapSort.out
Input numbers to sort:
225 129 82 957 785 281 623 871 519 539 939 125 769 226 58 147 210 124 82 339 869
290 446 226 103 341 473 474 683 199 426 177 794 78 781 695 369 537 22 40 145 53
5 275 223 203 939 300 894 25 287 847 981 176 329 134 70 44 423 219 510 45 615 49
7 794 334 993 564 908 468 884 656 294 224 289 639 378 598 976 505 806 918 955 21
8 96 335 574 652 97 992 862 883 235 513 344 843 435 755 581 719 -1

Result:
993 992 981 976 957 955 939 939 918 908 894 884 883 871 869 862 847 843 806 794
794 785 781 769 755 719 695 683 656 652 639 623 615 598 581 574 564 539 537 535
519 513 510 505 497 474 473 468 446 435 426 423 378 369 344 341 339 335 334 329
300 294 290 289 287 281 275 235 226 226 225 224 223 219 218 210 203 199 177 176
147 145 134 129 125 124 103 97 96 82 82 78 70 58 45 44 40 25 22
sp2019@sp2019-VirtualBox:~/algorithm$ diff -s output.txt test1.txt
Files output.txt and test1.txt are identical
```

(output.txt: Result로 만든 txt 파일, test1.txt: 엑셀을 통해 입력을 정렬한 txt파일, 둘의 내용이 동일한지 비교하여 정렬이 잘 되었는지 확인)

- n=1일 때

```
sp2019@sp2019-VirtualBox:~/algorithm$ ./1-1HeapSort.out
Input numbers to sort:
2 -1

Result:
2
```

1-2. 연습문제 6-5.9

※ 파일 이름: 1-2Practice.cpp, 1-2Practice.out

1) 헤더 파일 목록

- <iostream>, <fstream>, <sstream>

2) Struct

```
- struct nodestruct{
    int num;          //보관할 숫자
    int xaddress;     //해당 숫자가 있던 array의 x좌표
    int yaddress;     //해당 숫자가 있던 array의 y좌표
};
- struct HeapStruct{
    int size;         //Heap에 들어있는 element 수
    node *list;       //Heap
};
- node: struct nodestruct의 포인터
- MinHeap: struct HeapStruct의 포인터
```

3) Functions

function	설명
Parent(int x): int	index가 x인 node의 부모 index를 찾아서 return
LChild(int x): int	index가 x인 node의 왼쪽 자식 index를 찾아서 return
RChild(int x): int	index가 x인 node의 오른쪽 자식 index를 찾아서 return
swap(int* a, int* b): void	a와 b의 값을 서로 바꾼다.
BuildHeap(int size, MinHeap H): MinHeap	Heap을 만들고 Heap 안에 size만큼의 빈 node 포인터를 만든다. Heap의 size(들어 있는 element 수)는 0으로 초기화한다. 새로 만든 Heap을 return한다.
Min_Heapify(int nodenum, MinHeap H): void	index가 nodenum인 node가 Heap의 규칙에 알맞도록 array를 정렬한다.
Insert(node X, MinHeap H): void	node X를 MinHeap H에 입력하고 Heap의 규칙에 알맞도록 정렬한다. H의 size를 하나 키운다.
Delete_Min(MinHeap H): node	MinHeap H에서 제일 작은 node인 root 노드를 빼낸 뒤 Heap의 규칙에 알맞도록 정렬한다. H의 size를 하나 줄인다. 빼낸 노드를 return한다.

4) main 함수 동작 과정

- ① input1-2.txt 파일을 통해 k개의 배열을 입력 받는다.
- ② 각 배열의 첫 번째 원소를 Heap에 넣고 정렬한다.
- ③ Heap의 가장 첫 노드를 꺼내어 값을 출력하고 Heap을 다시 정렬한다.
- ④ 꺼낸 노드가 있던 배열에서 다음 노드를 Heap에 넣고 정렬한다.
- ⑤ 모든 배열의 원소를 Heap에 넣을 때까지 ③, ④ 과정을 반복한다.
- ⑥ Heap이 빌 때까지 Heap의 가장 첫 노드를 꺼내어 값을 출력하고 Heap을 다시 정렬한다.

5) 실행 화면 캡처

- 주어진 예시

```
sp2019@sp2019-VirtualBox:~/algorithm$ g++ -o 1-2Practice.out 1-2Practice.cpp
sp2019@sp2019-VirtualBox:~/algorithm$ cat input1-2.txt
4
0 2 3 5 5 10 30
0 5 6 7 10 13 15
1 7 11
17 18 21 23
sp2019@sp2019-VirtualBox:~/algorithm$ ./1-2Practice.out
0 0 1 2 3 5 5 5 6 7 7 10 10 11 13 15 17 18 21 23 30
```

- 각 배열의 원소 개수가 99개, $k=99$ 일 때

```

99
sp2019sp2019-VirtualBox-~/.algorithms8.txt cat input1-2.txt
sp2019sp2019-VirtualBox-~/.algorithms8.txt cat input1-2.txt
9 16 20 61 73 78 80 111 113 128 146 150 181 188 190 202 219 224 226 238 239 261 273 275 296 303 303 349 355 355 360 364 365 371 376 376 383 399 404 434 453 453 458 461 462 468 480 484 487 493 5
83 586 510 513 535 543 583 564 572 578 588 595 606 614 632 643 655 669 682 714 720 721 727 731 732 741 757 763 803 822 868 876 878 885 907 910 923 924 929 935 941 946 952 978 978 991
8 36 44 51 54 59 104 123 129 136 147 151 157 185 189 206 213 239 248 256 265 295 296 300 301 309 325 330 342 345 349 359 367 377 380 390 406 412 415 420 428 433 441 459 471 476 485 492
10 29 36 46 47 49 55 64 77 77 85 87 91 92 102 112 113 117 119 123 142 146 158 160 168 163 168 175 182 185 189 208 224 228 230 231 258 262 273 281 285 307 345 347 361 376 379 381 403 410 423 433 438
439 442 448 451 452 468 466 488 517 522 561 576 586 594 599 611 625 627 645 653 671 677 735 740 749 768 776 783 808 802 809 817 822 830 864 865 875 885 889 904 915 925 934 938 941 985 985 985
10 30 31 74 75 80 82 84 85 86 184 188 191 202 205 226 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 589 617 633 637 648 668 671 682 683 714 724 728 740 779 781 795 815 828 836 840 861 881 895 907 914 916 933 935 947 954 955 959 964 984 985 994
51 71 73 80 82 106 128 144 162 169 175 187 189 220 223 262 266 266 277 281 285 287 301 302 303 341 342 343 346 347 352 355 366 377 388 405 407 410 411 427 433 443 445 446 453 463 468 472 472 484 4
489 502 504 509 524 535 539 548 551 554 560 572 584 600 609 633 641 663 670 721 728 737 740 750 752 760 768 773 784 803 804 807 813 819 850 861 864 868 914 915 920 924 934 939 941 942 950 992
51 29 36 46 47 49 55 64 77 77 85 87 91 92 102 112 113 117 119 123 142 146 158 160 168 163 168 175 182 185 189 208 224 228 230 231 258 262 273 281 285 307 345 347 361 376 379 381 403 410 423 433 438
57 465 472 504 540 547 556 551 572 575 581 599 609 620 654 664 669 703 718 750 777 779 802 802 780 875 881 885 899 905 916 934 934 954 962 963 975 975 982
17 27 38 41 48 53 54 57 75 88 98 113 134 138 142 172 172 180 222 231 235 241 243 247 285 316 322 324 330 333 334 341 354 358 367 385 389 395 409 418 432 446 472 475 481 499 506 518 521 524
536 536 539 595 087 636 635 652 682 688 696 701 702 709 789 710 731 746 754 758 761 797 779 796 803 804 806 815 819 831 844 861 868 890 898 898 910 918 932 933 958 989 992 999
10 30 31 39 54 58 70 72 80 85 94 130 140 212 224 226 266 283 287 302 316 317 329 331 334 351 356 393 412 423 425 445 456 458 472 481 485 505 506 507 509 509 516 519 523 540 564 565 571 588 6
10 616 621 629 634 646 67 68 69 70 70 78 78 78 727 739 763 775 778 781 798 812 817 819 822 826 834 836 857 859 868 877 881 887 888 903 908 916 933 9
```

Input1-2.txt 로 들어간 파일(너무 길어서 일부 생략)

