

컴퓨터알고리즘과실습 실습4

2020112046 이재원

1번

```
#include <iostream>
using namespace std;

class PQ {
private:
    int N;
    int i;

public:
    int *a;
    int heapCompare;

    PQ(int max) {
        a = new int[max];
        N = 0;
        i = 0;
        heapCompare = 0;
    }
    ~PQ() {
        delete a;
    }

    void insertData(int v) {
        a[i++] = v;
    }

    int get(int i) {
        return a[i];
    }

    // insert a new element into the priority queue - O(logN)
    // !-----
    void MakeHeap(int a[], int root, int LastNode) {
        int child;
        int temp;
        temp = a[root];
        child = 2 * root;
        while (child <= LastNode) {
            heapCompare++;
            if ((child < LastNode) && (a[child] < a[child + 1])) {
                child++;
            }
            heapCompare++;
            if (temp >= a[child]) {
                break;
            }
        }
    }
};
```

```

        } else {
            a[child / 2] = a[child];
            child *= 2;
        }
    }
    a[child / 2] = temp;
}
};

void swap(int &a, int &b) {
    int temp = a;
    a = b;
    b = temp;
}

void heapSort(int a[], int N, PQ &pq) {
    for (int i = N / 2 - 1; i >= 0; i--)
        pq.MakeHeap(a, i, N - 1);
    for (int i = N - 1; i >= 1; i--) {
        swap(a[0], a[i]);
        pq.MakeHeap(a, 0, i - 1);
    }
}

int main() {
    int N;
    cin >> N;
    PQ pq(N);
    for (int i = 0; i < N; i++) {
        pq.insertData(rand() % 10000 + 1);
    }
    heapSort(pq.a, N, pq);
    cout << "Random Numbers Array - Compare count: " << pq.heapCompare << endl;

    //!-----
    //최악의 경우 : 이미 정렬된 배열을 입력받을 경우 -> 최대 힙을 사용하기 때문에 비교 횟수
    가 많아짐.
    int K = 10000;
    PQ pq2(K);
    for (int i = 0; i < K; i++) {
        pq2.insertData(i + 1);
    }
    heapSort(pq2.a, K, pq2);
    cout << "Sorted Array - Compare count: " << pq2.heapCompare << endl;
    //!-----
    for (int i = 0; i < N; i++) {
        cout << pq.get(i) << " ";
    }

    return 0;
}

```

실행 결과

```
> cd "/Users/jaewonlee/Desktop/PS/" && g++ -std=c++2a -g test.cpp -o a.out && "/Users/jaewonlee/Desktop/PS"/a.out
10000
Random Numbers Array - Compare count: 255182
Sorted Array - Compare count: 264526
1 1 3 3 5 5 6 6 7 7 8 9 9 10 11 11 12 13 15 15 16 17 17 19 20 20 21 23 26 27 28 30 31 31 32 34 34 34 34 38 38 40 41 41 41 41 42 42 43 46 48 49 50 51 52
56 58 59 60 60 60 63 65 66 66 66 67 68 69 69 70 71 72 74 75 77 78 79 79 79 79 79 81 81 83 86 87 89 90 94 95 95 95 95 95 97 101 102 102 106 107 107
8 109 109 109 110 111 113 114 115 115 116 116 118 118 118 118 119 120 121 122 125 126 127 128 129 129 129 130 130 134 137 138 138 141 141 142 142 143 14
144 144 145 146 147 148 148 149 149 151 151 152 153 153 154 156 157 159 160 161 161 161 162 163 163 164 164 166 168 168 171 172 172 172 173 174 174 17
176 177 177 178 179 181 182 183 183 183 185 186 188 188 191 192 193 194 195 195 195 197 197 198 198 199 199 201 203 204 204 204 205 205 205 205 206 206
10 211 212 212 215 215 216 218 219 219 219 220 220 220 222 222 224 227 227 230 231 231 232 232 232 234 234 235 235 237 237 239 241 241 243 243 243
7 249 249 251 252 254 255 257 257 258 258 259 261 261 262 262 263 263 265 265 265 266 266 267 268 269 270 271 271 273 274 275 277 277 279 280 281 283 2
286 290 294 295 295 296 296 296 296 297 297 299 300 300 303 306 306 307 308 311 313 313 317 318 318 319 319 319 320 320 321 321 322 324 325 329 331 33
334 334 336 338 338 339 340 341 341 342 342 343 344 344 348 351 356 357 357 358 363 363 364 364 364 365 365 366 367 367 368 370 372 372 372 372 373 374
77 378 379 380 381 382 382 382 385 385 385 386 387 388 388 388 391 391 392 393 393 395 395 396 397 397 398 400 402 404 405 406 408 409 411 412 413 413
4 415 416 416 416 417 417 418 418 418 418 419 419 420 420 421 421 421 422 424 424 424 435 435 436 436 438 441 442 443 444 445 445 447 447 447 447 448 4
450 451 451 452 457 458 458 459 461 461 462 463 464 467 471 471 471 475 476 477 478 481 483 484 485 485 486 488 489 490 491 492 492 493 493 493 497 49
501 501 503 504 505 505 506 509 511 512 513 513 514 515 517 517 517 518 520 520 521 523 523 523 524 524 524 525 525 526 527 528 531 531 531 533 535 535
38 538 540 540 541 543 546 546 546 546 546 547 547 548 548 548 550 550 552 553 554 554 556 557 557 558 558 559 559 560 561 561 562 562 563 563 563
7 568 569 569 570 570 572 572 576 578 578 579 580 581 583 583 585 587 588 588 589 590 590 590 591 591 592 593 594 594 596 597 598 598 599 601 6
604 604 606 606 606 607 607 608 615 616 620 621 624 624 624 625 625 625 626 626 629 629 629 630 633 633 634 634 634 634 635 636 638 640 640 64
643 643 643 644 646 647 649 649 651 656 656 656 656 657 657 657 660 661 661 662 662 662 663 666 668 670 671 673 674 674 675 675 676 676 677 678 679 680
```

heapSort가 최악의 시간복잡도를 겪을 때는, 이미 정렬된 값을 넣었을 때 이므로, 정렬하여 넣어 주었다.

그러나 heapSort는 최선, 최고, 최악 모두 time complexity가 $O(n\log n)$ 이다.

비교 횟수가 그리 크지 않은 차이를 보이는 것을 알 수 있다.

2번

```
#include <iostream>
using namespace std;
```

```
int compareCount = 0;
```

```
void swap(int a[], int i, int j) {
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}
```

```
int partition(int a[], int l, int r) {
    int i, j;
    int v;
```

```
    if (r > l) {
        v = a[l];
        i = l;
        j = r + 1;
        while (1) {
            while (a[++i] < v)
                if (i == r) break;
            while (a[--j] > v)
                if (j == l) break;
            compareCount++;
        }
    }
}
```

```

        if (i >= j) break;
        swap(a, i, j);
    }
    swap(a, l, j);
}
return j;
}

void select(int a[], int l, int r, int k) {
    int j;
    if (r > l) {
        j = partition(a, l, r);
        if (j > l + k - 1) select(a, l, j - 1, k);
        if (j < l + k - 1) select(a, j + 1, r, k - (j - l + 1));
    }
}

int main() {
    int N;
    cin >> N;
    int a[N], b[N], c[N];
    for (int i = 0; i < N; i++) {
        a[i] = rand() % 10000 + 1;
        b[i] = a[i];
        c[i] = a[i];
    }

    select(a, 0, N - 1, 1);
    cout << "Minimum: " << a[0] << ", Compare: " << compareCount << endl;
    compareCount = 0;
    select(b, 0, N - 1, N);
    cout << "Maximum: " << b[N - 1] << ", Compare: " << compareCount << endl;
    compareCount = 0;
    select(c, 0, N - 1, N / 2);
    cout << "Middle: " << c[N / 2] << ", Compare: " << compareCount << endl;

    // print a array
    for (int i = 0; i < N; i++) {
        cout << a[i] << " ";
    }
}

```

실행 결과

```
● > cd "/Users/jaewonlee/Desktop/PS/" && g++ -std=c++2a -g test.cpp -o
10000
Minimum: 1, Compare: 3145
Maximum: 10000, Compare: 3184
Middle: 4927, Compare: 6541
1 1 7 16 20 13 7 3 6 8 17 19 15 15 5 5 11 10 9 17 9 11 3 12 6 20 130
118 95 122 78 118 66 95 126 109 58 86 97 109 146 134 147 110 60 79
9 130 60 90 52 116 102 107 42 51 138 27 21 107 138 129 50 30 142 144
48 102 70 145 38 141 148 243 154 257 206 183 210 168 249 157 243 160
178 237 176 247 205 246 201 232 174 162 172 216 183 171 188 159 156
66 173 182 198 179 172 204 203 210 220 186 257 175 234 194 185 258 2
1 210 187 210 211 227 151 152 205 210 185 222 224 201 222 262 222 26
```

Minimum, Maximum값을 찾을 때는 partition한 후 한 쪽만 보면 되기 때문에, Middle값을 찾을 때보다 2배정도 비교 횟수가 작아지는 것을 볼 수 있으며, Minimum, Maximum값을 찾는 비교 횟수는 비슷한 것을 볼 수 있다.

3번

```
#include <algorithm>
#include <fstream>
#include <iostream>
#include <map>
#include <sstream>
#include <string>
#include <vector>

using namespace std;
map<string, int> word;
map<string, int>::iterator iter;

int main() {
    string line;
    string buffer;

    ifstream file("datafile.txt");
    if (file.is_open()) {
        while (getline(file, line)) {
            // split the line into words with blank
            istringstream iss(line);
            while (getline(iss, buffer, ' ')) {
                // remove , and " and . and : and ; and ? and ! from the buffer
                buffer.erase(remove(buffer.begin(), buffer.end(), ','), buffer.end());
                buffer.erase(remove(buffer.begin(), buffer.end(), '"'), buffer.end());
                buffer.erase(remove(buffer.begin(), buffer.end(), '.'), buffer.end());
                buffer.erase(remove(buffer.begin(), buffer.end(), ':'), buffer.end());
                buffer.erase(remove(buffer.begin(), buffer.end(), ';'), buffer.end());
                buffer.erase(remove(buffer.begin(), buffer.end(), '?'), buffer.end());
                buffer.erase(remove(buffer.begin(), buffer.end(), '!'), buffer.end());
                // if the word is not in the map, insert it
                if (word.find(buffer) == word.end()) {
                    word[buffer] = 1;
                } else {
                    word[buffer]++;
                }
            }
            // cout << line << endl;
        }
        file.close();
    } else {
        cout << "Unable to open file";
        return 1;
    }
}
```

```

    // print the result
    for (iter = word.begin(); iter != word.end(); iter++) {
        cout << iter->first << " " << iter->second << endl;
    }
    return 0;
}

```

실행 결과

```

test.cpp x
test.cpp > main()
31 // if the word is not in the map, insert it
32 if (word.find(buffer) == word.end()) {
33     word[buffer] = 1;
34 } else {
35     word[buffer]++;
36 }
37 // cout << line << endl;
38 }
39 file.close();
40 } else {
41     cout << "Unable to open file";
42     return 1;
43 }
44 }
45
46 // print the result
47 for (iter = word.begin(); iter != word.end(); iter++) {
48     cout << iter->first << " " << iter->second << endl;
49 }
50 return 0;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

amazement 1
among 3
an 11
and 47
animal 1
another 10
answer 1
answered 5
any 14
anybody 1
anyone 2
anything 5
apart 1
apparition 1
are 13
arithmetic 2
around 1
aroused 1
as 20
aside 1
ask 3
asked 6
assistance 1
asteroid 4
astonishment 1
astounded 1
astronomer 4
at 14
attempt 1

```

어린왕자(Little Prince) 의 일부분을 발췌하여 실행해봤음.

소감

슬슬 재밌는 것 같습니다.. ^o^