

데이터 구조 설계

Project #2

제출일: 2023-11-15

학과: 컴퓨터정보공학부

학번: 2020202034

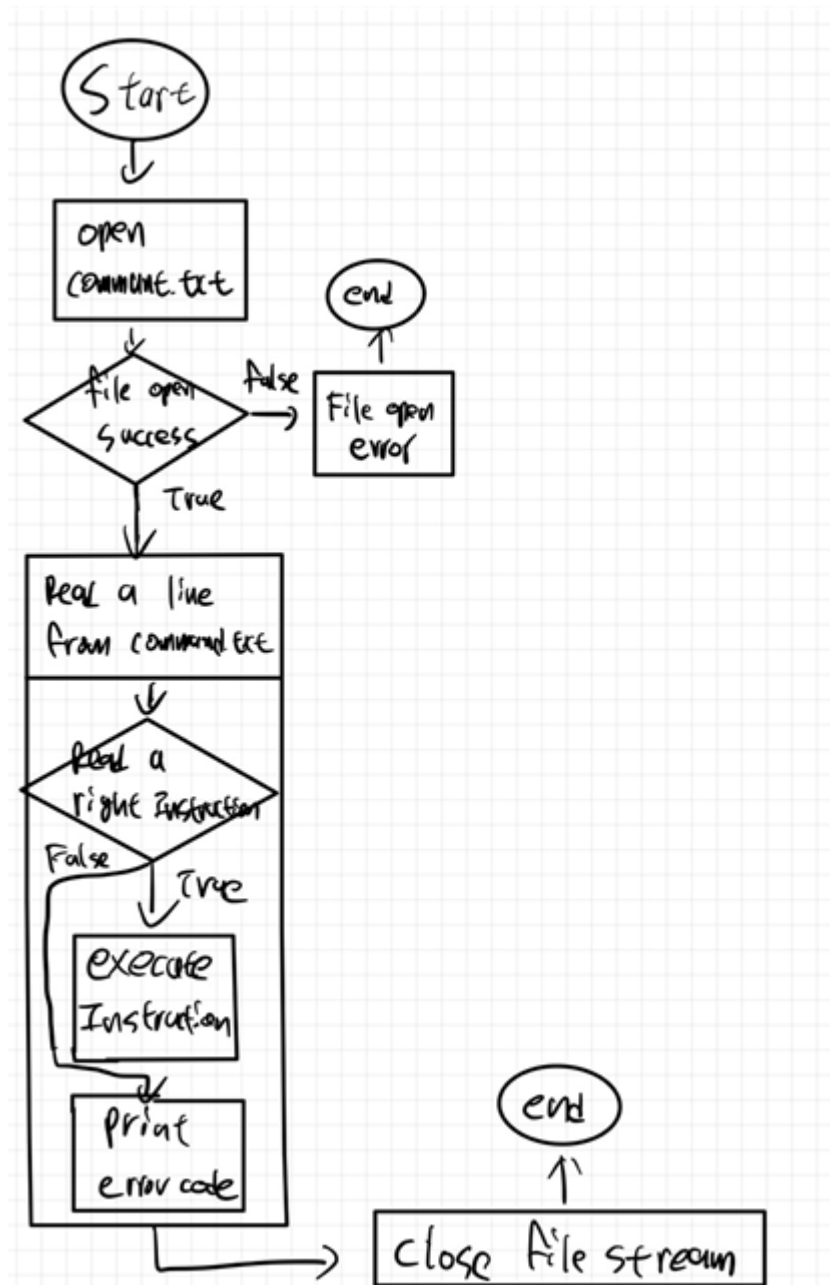
이름: 김태완

1. Introduction

- A. 과제 설명: B+-Tree, Selection Tree, Heap 자료구조를 이용해 도서 대출 관리 프로그램을 만드는 과제입니다. 대출 가능한 도서는 B+-Tree로 관리하고, 대출 불가능한 도서는 Selection Tree와 Heap으로 관리합니다. B+-Tree의 Index Node의 key는 책 제목이고, Data Node는 Key를 책 제목으로 가지고, Value로 책 정보를 담은 객체의 주소를 가집니다. Selection Tree는 run으로 각 도서 분류 코드별로 구분된 heap 8개를 가지고(해당 도서 분류 코드의 대출 불가 도서가 존재하지 않으면 heap이 없을 수도 있음) 책 제목을 key로 가지며 winner tree의 구조를 가집니다. 프로그램은 Command.txt 파일에 적힌 명령어를 한 줄씩 읽으며 명령어에 따른 내용을 수행하게 됩니다. 명령어는 LOAD, ADD, SEARCH_BP, PRINT_BP, PRINT_ST, DELETE, EXIT이 있습니다.
- B. LOAD: loan_book.txt 파일에 작성된 책의 정보를 읽어와 B+-Tree에 저장합니다. 책 정보는 책 제목, 도서 코드, 저자, 발행 연도, 대출된 책 권수로 이루어져 있으며, 'wt'으로 구분되어 있습니다(책 이름, 저자에 공백 문자 존재 가능). 해당 파일이 존재하지 않거나, B+-Tree가 차 있다면 에러 코드를 출력합니다.
- C. ADD: B+-Tree에 책 정보를 추가로 작성합니다. 책 제목, 도서 코드, 저자, 발행 연도 정보가 'wt'를 기준으로 구분되어 입력되며 기존에 해당 도서가 존재한다면 기존 도서 정보의 현재 대출 권수를 1 올려주고, 해당 도서 코드를 기준으로 도서가 모두 대출되었다면 해당 도서 정보를 Selection Tree로 넘겨줍니다. 도서가 존재하지 않았다면 대출 권수를 0으로 하고 B+-Tree에 정보를 추가합니다. 정보가 하나라도 빠져 있으면 에러 코드를 출력합니다.
- D. SEARCH_BP: 인자가 1개 들어오면 해당 인자를 제목으로 하는 책 정보를 찾아 출력합니다. 인자가 2개 들어오면 두 인자 사이 범위에 존재하는 책 제목을 가진 도서 정보를 모두 출력합니다. 인자의 개수가 1, 2개가 아니거나, B+-Tree가 비었거나, 해당 도서가 존재하지 않으면 에러 코드를 출력합니다.
- E. PRINT_BP: B+-Tree에 존재하는 도서 정보를 알파벳 순으로 전부 출력합니다. B+-Tree가 비어 있다면 에러 코드를 출력합니다.
- F. PRINT_ST: 인자로 도서 분류 코드를 받아, 해당 코드의 도서 정보를 가진 Heap의 도서 정보를 제목의 알파벳 순서대로 출력합니다.
- G. DELETE: Selection Tree에서 가장 이름 순으로 앞선 책 정보(Winner Tree의 winner)를 지우고 Selection Tree와 Heap의 정보를 재정렬합니다.
- H. EXIT: 프로그램을 종료합니다.

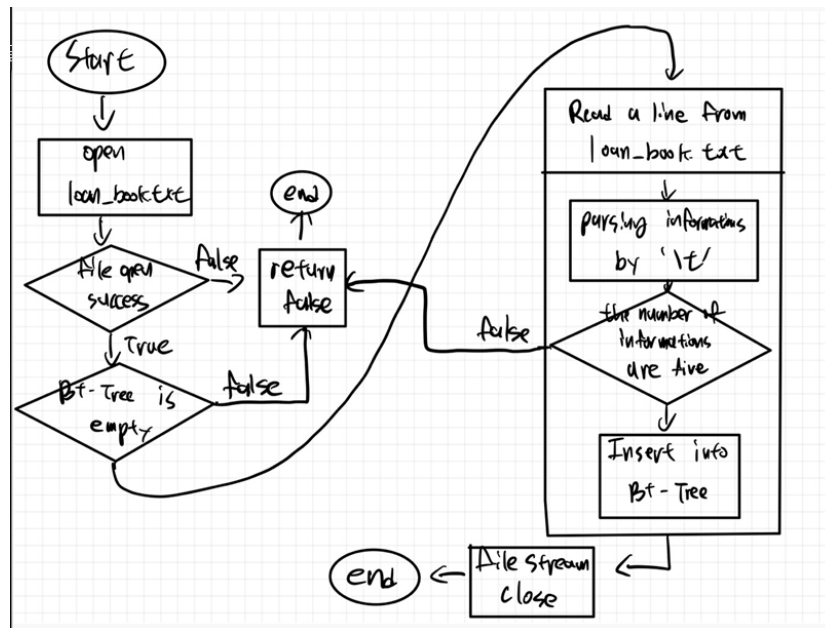
2. Flow Chart

A. RUN



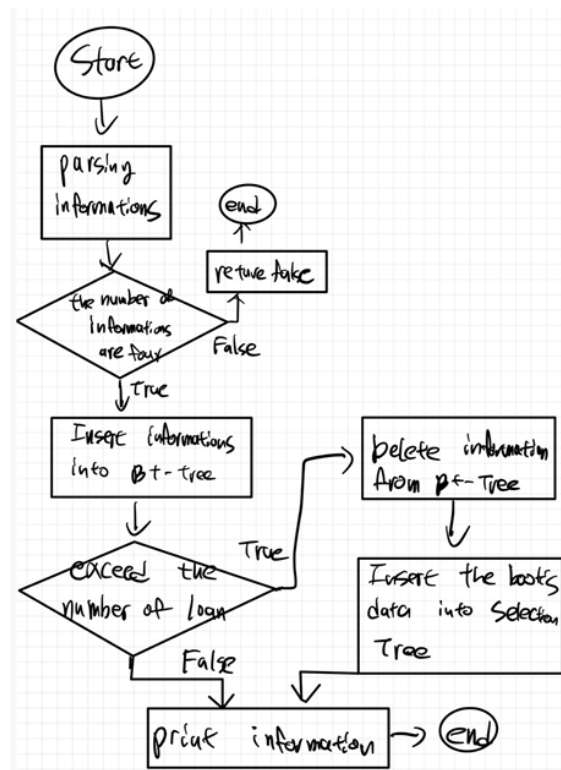
Main 함수 내에서 실행되는 run 함수의 flowchart입니다. 이 함수 내에서 명령어를 한 줄씩 읽어 해당하는 명령어 함수를 호출합니다.

B. LOAD



LOAD 명령어의 flowchart입니다. Loan_book.txt 파일 스트림을 열어 파일을 한 줄씩 읽으며 데이터를 B+-Tree에 삽입합니다.

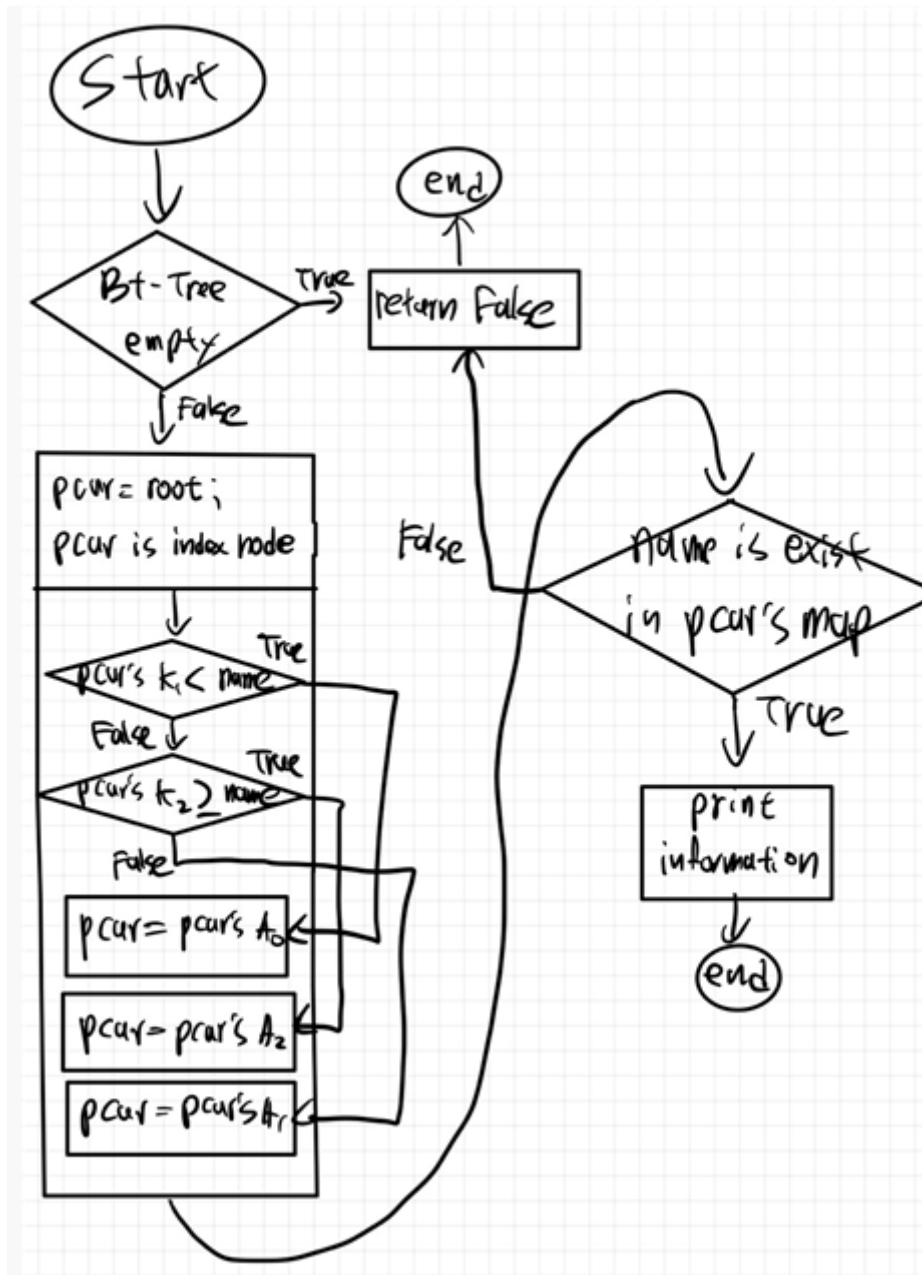
C. ADD



ADD 명령어의 flowchart입니다. 데이터를 B+-Tree에 추가적으로 넣거나 기존 데이터의 대출 권수를 올려주며, 대출 권수가 초과하면 Selection Tree로 넘겨주는 기능을 수행합니다.

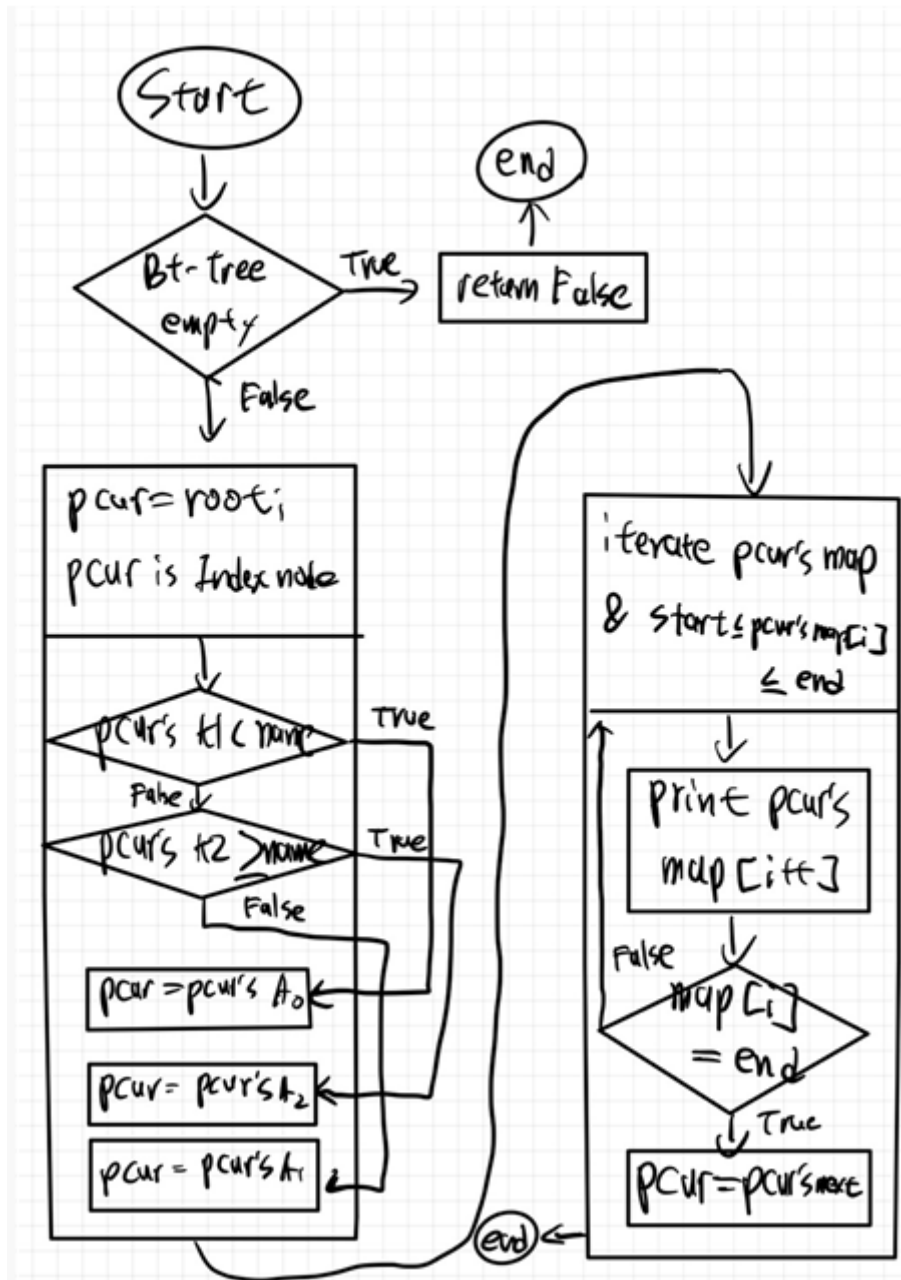
D. SEARCH_BP

i. SEARCH_BP name



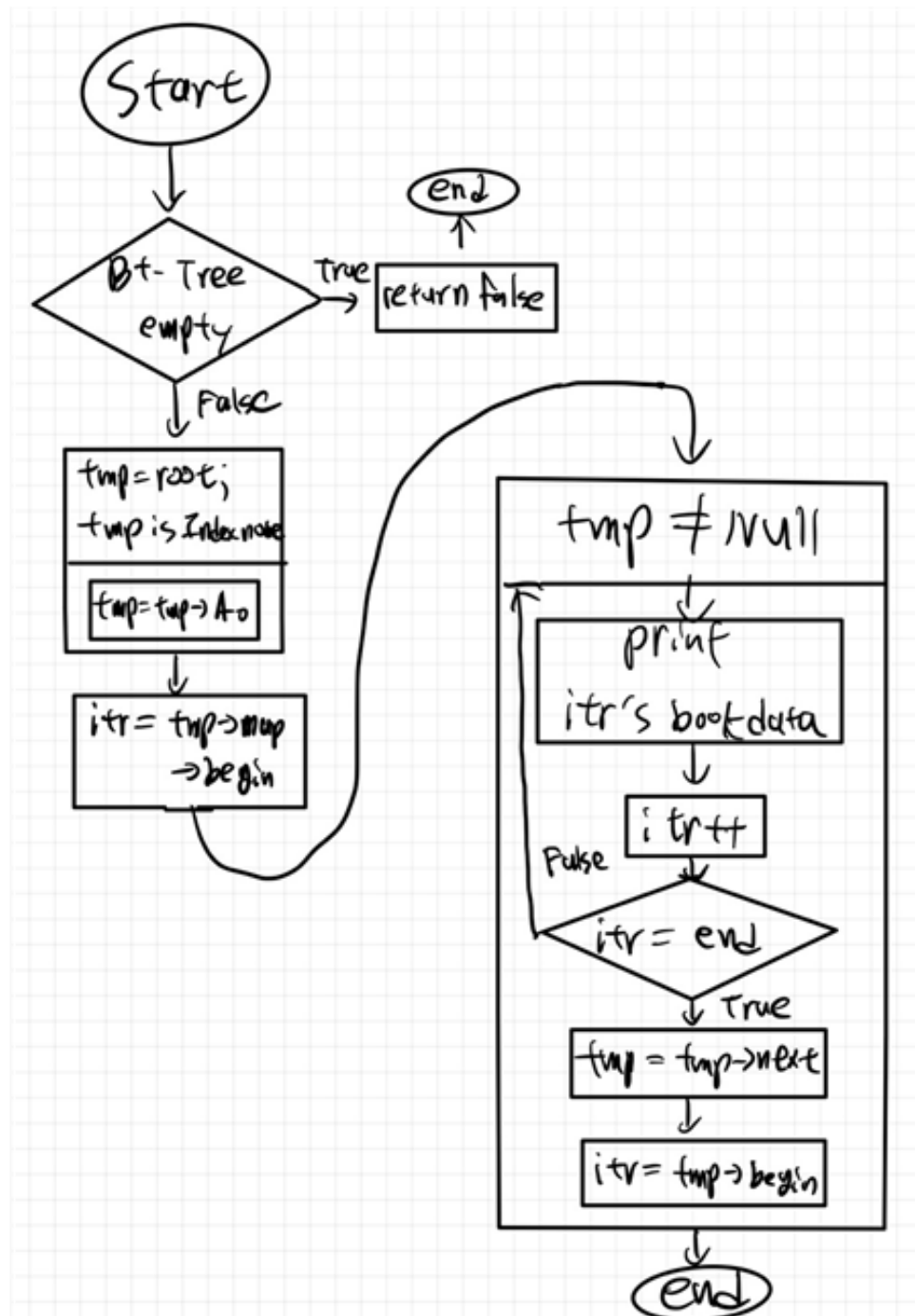
SEARCH_BP 명령어 중 인자를 한 개 받는 경우의 flowchart입니다. 검색할 이름을 받아 B+-Tree 에서 찾고, 존재하면 해당 정보를 출력합니다.

ii. SEARCH_BP range_1 range_2



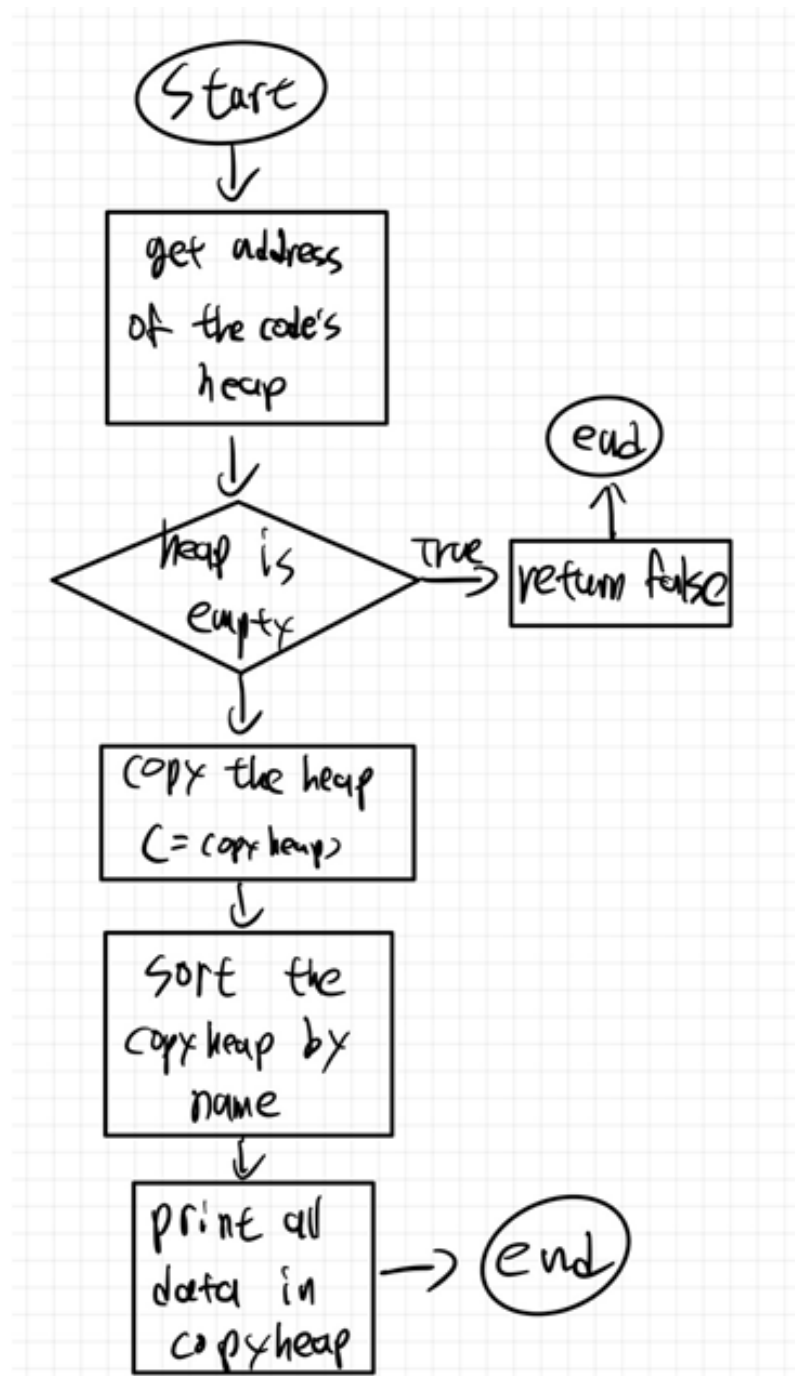
SEARCH_BP 명령어 중 인자가 2개 들어왔을 때의 flowchart입니다. 범위 내에 들어오는 데이터를 전부 출력하고, 출력할 게 없다면 false를 반환합니다.

E. PRINT_BP



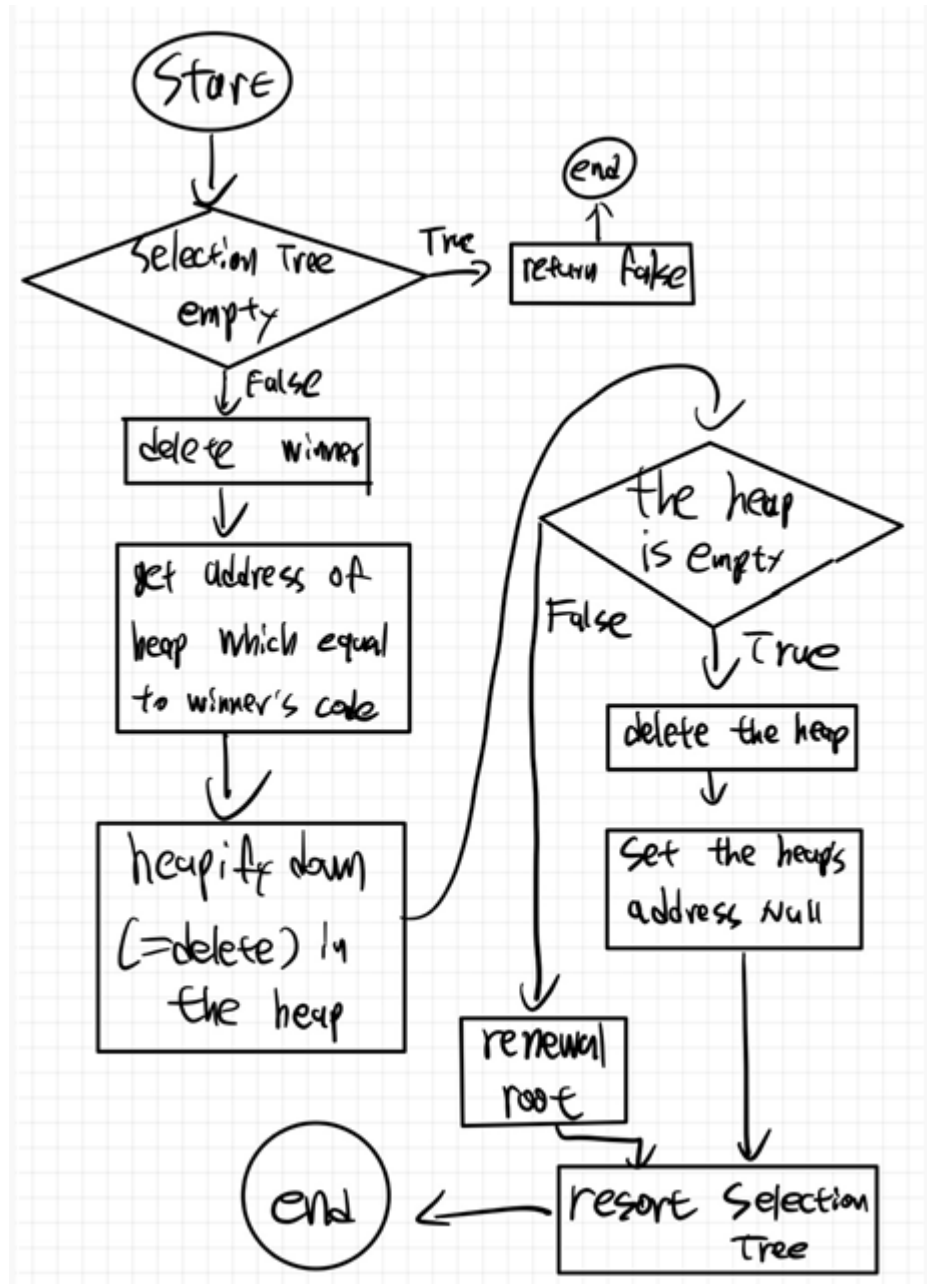
PRINT_BP의 flowchart입니다. B+-Tree의 root index node부터 MostLeftChild를 계속 따라 내려가며 맨 앞 데이터부터 순서대로 출력합니다.

F. PRINT_ST



PRINT_ST의 flowchart입니다. 인자로 받은 코드의 힙을 제목 순으로 출력합니다.

G. DELETE



DELETE의 flowchart입니다. Selection Tree의 최상단 winner를 지우고, 이를 heap에도 반영하며, 바뀐 heap의 정보로 selection tree도 resort합니다.

3. Algorithm

A. B+-Tree

- i. B+-Tree는 B-Tree를 개선한 것으로 Index node와 Data node로 나뉩니다. 트리 말단에 존재하는 Data node에만 정보가 저장되고, Index node에는 다른 노드의 주소와 인덱스(키 값)만 존재합니다. 탐색 시에 B-Tree의 경우 해당 키 값을

찾으면 그 위치에서 바로 데이터를 반환할 수 있지만 B+-Tree는 키 값을 찾아도 Data Node가 있는 말단까지 내려가야 데이터를 반환할 수 있기 때문에 탐색 시간이 더 걸리지만, B+-Tree는 데이터 노드끼리 연결되어 있어 범위 탐색에 용이하다는 장점이 있습니다.

ii. Insert in B+-Tree

1. B+-Tree가 비었다면, 노드 하나 삽입하고 종료
2. Root index node부터 다음의 기준으로 data node까지 내려감
 - A. 입력하는 데이터의 키 값이 현재 index node의 K_1 보다 작으면 index node를 A_0 으로 이동시킴
 - B. 키 값이 $K_n \leq \text{Key} < K_{n+1}$ 이면 A_n 으로 index node를 이동시킴
 - C. 이때, $K_{n+1} = \text{inf}$ 라고 가정
3. 해당 키 값을 가진 데이터가 데이터 노드에 존재한다면 데이터 교체
4. 없다면, 데이터 노드에 새 데이터 삽입
5. 만약 B+-Tree의 차수 - 1보다 데이터 노드의 데이터 수가 많으면 가운데 데이터의 키 값을 상위 인덱스 노드로 추가하고, 가운데 데이터보다 키 값이 크거나 같은 노드를 새 데이터 노드로 분리 후 인덱스 노드, 데이터 노드 상호 연결
6. 인덱스가 하나 추가된 상위 인덱스 노드도 차수 - 1 보다 인덱스가 더 많다면, 가운데 인덱스를 상위 인덱스 노드로 올리고, 인덱스 노드를 가운데 인덱스를 기준으로 쪼개고 모두 연결시킵니다. 그리고 상위 인덱스 노드의 인덱스 수가 차수 - 1과 같거나 적을 때까지 이를 반복합니다.

B. Selection Tree & Heap

- i. Selection Tree(Winner Tree)는 데이터들을 보관한 run을 말단 노드로 가지고 아래에서부터 데이터들의 키 값을 비교하며 더 우선순위를 가지는 데이터를 위로 올리며 최종적으로 root에 가장 우선순위를 가지는 run의 데이터 하나를 가지게 되는 자료 구조입니다. 이번 프로젝트에서는 run의 개수가 8개로 정해져 있어 초기에 이를 바탕으로 selection tree를 구성해 heap에 데이터가 추가되거나 delete가 일어날 때마다 selection tree를 재구성하여 구조를 유지합니다. Heap은 전체 트리에서 부모 노드의 우선 순위가 자식 노드보다 높은 트리 구조를 말합니다. Heap은 기본적으로 complete tree의 구조를 가져 데이터를 추가할 때 가장 하위 level에서 가장 왼쪽에 데이터를 추가합니다.

ii. Resort in Selection Tree

1. Heap에서 insert 또는 delete가 이루어져 root가 변경될 수 있을 때 이 과정을 진행하는 함수가 호출됨
2. Root의 변화가 있을 수 있는 run을 가진 selection tree의 말단 노드에서 부터 시작
3. 현재 노드와 같은 부모를 가진 형제 노드를 찾습니다.
4. 둘 다 데이터를 가지지 않는다면 부모 노드에 NULL을 전달하고, 둘 중 하나만 가지지 않는다면 NULL이 아닌 데이터를 부모에 전달합니다. 둘 다 데이터를 가진다면 둘 중에 키 값이 더 우선순위를 가지는 데이터를 부모 노드로 올립니다.
5. 현재 노드가 root 노드일 때까지 위 과정을 반복합니다.

iii. Heapify up(insert)

1. Heap은 complete tree 상에서 가장 마지막 노드의 위치에 새 노드를 삽입하는데, 이는 level order로 기존의 heap을 순회하고, 처음으로 NULL이 되는, 즉 새 노드를 추가할 위치의 부모 노드를 찾아 해당 위치에 새 노드를 삽입하면 됩니다.
2. 그리고 heapify up을 새 노드부터 시작합니다.
3. (heapify up) 현재 노드가 root이거나 현재 노드의 키 값보다 부모 노드의 키 값이 우선순위를 가지면 아래 과정을 종료합니다.
4. 부모 노드와 현재 노드의 데이터를 swap합니다.
5. 부모 노드에서 heapify up을 반복합니다.

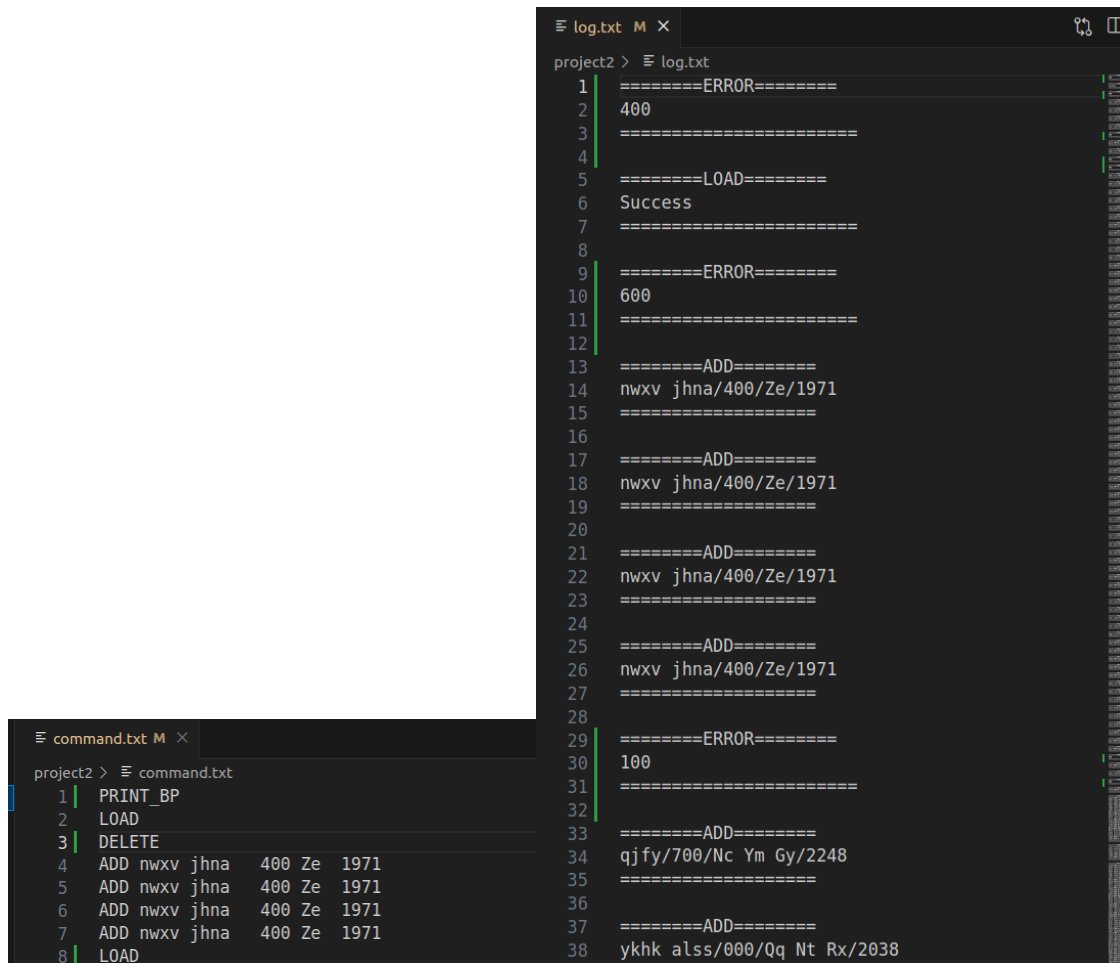
iv. Heapify down(delete)

1. heap에서 delete는 root node에서만 이루어집니다. 이때 마지막 노드와 root node의 데이터를 교환하고 마지막 노드를 삭제합니다(자식이 없기 때문에 root에 비해 삭제가 편함). 그리고 root에서부터 heapify down을 수행합니다.
2. (heapify down) 현재 노드가 root라면 마지막 노드를 찾아 데이터를 교환합니다(이때, root만 존재한다면 그냥 root를 지우고 heap을 비움.). 그리고 마지막 노드를 지웁니다.
3. 현재 노드의 왼쪽 자식, 오른쪽 자식 중 존재하면서 키 값이 더 적은 우

선순위를 가지는 노드를 찾습니다. 이때 자식이 없으면 heapify down을 종료합니다.

4. 전 과정에서 찾은 자식과 키 값을 비교해 현재 노드가 우선순위를 가지면 종료하고, 그렇지 않다면 데이터를 교환하고 해당 자식 노드에서 heapify down을 반복합니다.

4. Result Screen



```
project2 > command.txt
1 | PRINT_BP
2 | LOAD
3 | DELETE
4 | ADD nwxv jhna 400 Ze 1971
5 | ADD nwxv jhna 400 Ze 1971
6 | ADD nwxv jhna 400 Ze 1971
7 | ADD nwxv jhna 400 Ze 1971
8 | LOAD

project2 > log.txt
1 | =====ERROR=====
2 | 400
3 | =====
4 |
5 | =====LOAD=====
6 | Success
7 | =====
8 |
9 | =====ERROR=====
10 | 600
11 | =====
12 |
13 | =====ADD=====
14 | nwxv jhna/400/Ze/1971
15 | =====
16 |
17 | =====ADD=====
18 | nwxv jhna/400/Ze/1971
19 | =====
20 |
21 | =====ADD=====
22 | nwxv jhna/400/Ze/1971
23 | =====
24 |
25 | =====ADD=====
26 | nwxv jhna/400/Ze/1971
27 | =====
28 |
29 | =====ERROR=====
30 | 100
31 | =====
32 |
33 | =====ADD=====
34 | qjfy/700/Nc Ym Gy/2248
35 | =====
36 |
37 | =====ADD=====
38 | ykhk alss/000/Qq Nt Rx/2038
```

왼쪽의 command.txt를 맨 윗줄부터 보면 LOAD 하기 이전에 PRINT_BP를 했으므로 에러코드 400을 출력한 것을 볼 수 있습니다. 그 후 LOAD를 해 성공한 것을 볼 수 있고, 그 다음에 DELETE를 했는데 ADD 명령어로 추가한 도서가 없기 때문에 Selection이 비어 있을 것이므로 에러 코드 600이 출력되었습니다. 그 후 ADD한 데이터가 정상적으로 ADD된 것을 볼 수 있고, 데이터를 LOAD하고 ADD한 이후에 다시 LOAD했을 때 에러코드 100이 출력된 것을 볼 수 있습니다.

```

9  ADD qjfy 700 Nc Ym Gy 2248
10 ADD ykhk alss 0 Qq Nt Rx 2038
11 ADD asdf sdf 100 asdfsd
12 PRINT_ST 200
13 ADD ykhk alss 0 Qq Nt Rx 2038
14 ADD ykhk alss 0 Qq Nt Rx 2038
15 ADD qayc vwjc 600 Vp Td Wx 1980
16 ADD qayc vwjc 600 Vp Td Wx 1980
17 ADD siyg bqew 500 Sx 1919
18 ADD siyg bqew 500 Sx 1919
19 ADD awvp 700 Sz 2185
20 ADD awvp 700 Sz 2185
21 ADD ywlx crqh 200 Wb 1986
22 ADD ywlx crqh 200 Wb 1986
23 ADD twdh zivi 700 Pt 2328
24 ADD fmvn 200 Ho 2010
25 ADD fmvn 200 Ho 2010
26 ADD fmvn 200 Ho 2010
27 ADD dapx 300 Sj Xa Nr 2241
28 ADD dapx 300 Sj Xa Nr 2241
29 ADD dapx 300 Sj Xa Nr 2241
30 ADD dapx 300 Sj Xa Nr 2241
31 ADD auby 500 Ii Ww 2201
32 ADD cluh iers 300 Ns Ow 2136
33 ADD cluh iers 300 Ns Ow 2136
34 ADD cluh iers 300 Ns Ow 2136
35 ADD cluh iers 300 Ns Ow 2136
36 ADD ixdg khsy 0 Av Te 2291
37 ADD ixdg khsy 0 Av Te 2291
38 ADD ixdg khsy 0 Av Te 2291

```

```

log.txt M x
project2 > log.txt
40
41 =====ERROR=====
42 200
43 =====
44
45 =====ERROR=====
46 500
47 =====
48
49 =====ADD=====
50 ykhk alss/000/Qq Nt Rx/2038
51 =====
52
53 =====ADD=====
54 ykhk alss/000/Qq Nt Rx/2038
55 =====
56
57 =====ADD=====
58 qayc vwjc/600/Vp Td Wx/1980
59 =====
60
61 =====ADD=====
62 qayc vwjc/600/Vp Td Wx/1980
63 =====
64
65 =====ADD=====
66 siyg bqew/500/Sx/1919
67 =====
68
69 =====ADD=====
70 siyg bqew/500/Sx/1919
71 =====
72
73 =====ADD=====
74 awvp/700/Sz/2185
75 =====
76

```

11번째 줄 명령을 보면 ADD 명령어의 인자가 asdf sdf, 100, asdfsd 총 3개이므로 에러 코드 200을 출력한 것을 볼 수 있습니다. 그 아래에 PRINT_ST 200 명령어를 실행하도록 했는데, 위 ADD한 내용을 보면 아직 코드가 200인 책이 대출 불가 도서로 변경되지 않았기 때문에 코드가 200인 대출 불가 도서를 저장하는 heap이 비어있으므로 에러코드 500이 출력되었습니다. 나머지 ADD도 성공적으로 이루어졌음을 볼 수 있습니다.

```
command.txt M X
project2 > command.txt
82 ADD glul 0 Xp Us 2285
83 ADD glul 0 Xp Us 2285
84 ADD how to make money 300 jh lee 2023
85 SEARCH_BP
86 SEARCH_BP tsmr
87 SEARCH_BP oosy xmsa
88 SEARCH_BP sdf sdf sdf
89 SEARCH_BP fsfb
90 SEARCH_BP how to make money
91 SEARCH_BP a i
92 SEARCH_BP y z
93 SEARCH_ST
94 PRINT_BP
95 PRINT_ST 000
96 PRINT_ST 100
97 PRINT_ST 200
98 PRINT_ST 300
99 PRINT_ST 400
100 PRINT_ST 500
101 PRINT_ST 600
102 PRINT_ST 700
103 DELETE
104 DELETE
105 DELETE
106 PRINT_ST 000
107 PRINT_ST 100
108 PRINT_ST 200
109 PRINT_ST 300
110 PRINT_ST 400
111 PRINT_ST 500
112 PRINT_ST 600
113 PRINT_ST 700
114 EXIT

log.txt M X
project2 > log.txt
332
333 =====ADD=====
334 how to make money/300/jh lee/2023
335 =====
336
337 =====ERROR=====
338 300
339 =====
340
341 =====SEARCH_BP=====
342 tsmr/000/Kz Vs/2016/1
343 =====
344
345 =====SEARCH_BP=====
346 oosy xmsa/200/Iz Yq/2271/1
347 =====
348
349 =====ERROR=====
350 300
351 =====
352
353 =====SEARCH_BP=====
354 fsfb/200/Mi/2357/1
355 =====
356
357 =====SEARCH_BP=====
358 how to make money/300/jh lee/2023/0
359 =====
360
```

위의 ADD한 내용은 기존에 존재하는 책을 추가하는 내용이었지만, "ADD how to make money ..." 부분은 새로운 책 데이터를 추가하는 부분입니다. 이 또한 성공적으로 되었음을 볼 수 있습니다. 그 아래에 SEARCH_BP로 새로 추가한 도서를 검색하니 잘 출력되었음을 볼 수 있습니다. 그리고 85번째 줄에 SEARCH_BP 명령어의 인자가 없고, 88번째 줄에는 인자가 3개 있으므로 둘 다 에러 코드 300을 출력했음을 볼 수 있습니다.

```
91 SEARCH_BP a i
92 SEARCH_BP y z
93 SEARCH_ST
```

```
log.txt M x
project2 > log.txt
361 =====SEARCH_BP=====
362 ajwp/400/In/1935/0
363 aoes dada/500/Fg Qo/2099/1
364 aulo sezk/100/Ms/2093/0
365 bumf pbbu/200/Sd Lq Cx/2294/0
366 cppl tzpe/000/Bf Gi Fe/2190/0
367 ctgh xbsa/300/Ap Nb Nu/2328/1
368 dzux kilw/000/Kq/2354/1
369 edvi/200/Aw/2083/1
370 emna/100/Ke/2112/0
371 enno rhzw/200/Mm/2158/0
372 eslx bguh/600/Fp Ml Aq/2067/0
373 ezyn/400/Sb/2363/0
374 fsfb/200/Mi/2357/1
375 fzqy/500/Bk Ui/2089/0
376 gbiy/200/Gg/1978/0
377 gnuq/000/Bg/2070/1
378 haln qfjz/500/Th Yb Po/2199/0
379 hieq zsnz/100/Zv Eb Eb/2318/1
380 how to make money/300/jh lee/2023/0
381 hxcj gljh/200/Px Wq/2379/1
382 igwd/200/Fr Qm/2110/1
383 iomc/400/Vc Bo Tt/2348/1
384 =====
385
386 =====SEARCH_BP=====
387 zcjm fiva/200/Su Mh/1926/1
388 zdvt qnfe/600/Wu Tz El/2264/0
389 zjep rucx/500/Up Rn Gz/2032/1
390 zjhg hmdm/600/Tr Vw/2010/1
391 zwag kkpw/500/Dq/2374/0
392 zxil/500/Tb Di Qw/2296/0
393 =====
394
395 =====ERROR=====
396 700
397 =====
398
```

SEARCH_BP 범위 검색 명령어를 입력하니 a로 시작하는 제목부터 i로 시작하는 제목의 책이 출력되었고, y~z로 시작하는 제목을 가지는 책 모두가 출력되었습니다. 그리고 SEARCH_ST라는 명령어는 없으므로 에러코드 700을 출력합니다.

```

399 =====PRINT_BP=====
400 ajwp/400/In/1935/0
401 aoes dada/500/Fg Qo/2099/1
402 aulo sezk/100/Ms/2093/0
403 bumf pbbu/200/Sd Lq Cx/2294/0
404 cppl tzpe/000/Bf Gi Fe/2190/0
405 ctgh xbsa/300/Ap Nb Nu/2328/1
406 dzux kilw/000/Kq/2354/1
407 edvi/200/Aw/2083/1
408 emna/100/Ke/2112/0
409 enno rhzw/200/Mm/2158/0
410 eslx bguh/600/Fp Ml Aq/2067/0
411 ezyn/400/Sb/2363/0
412 fsfb/200/Mi/2357/1
413 fzqy/500/Bk Ui/2089/0
414 gbiy/200/Gg/1978/0
415 gnuq/000/Bg/2070/1
416 haln qfjz/500/Th Yb Po/2199/0
417 hieq zsnz/100/Zv Eb Eb/2318/1
418 how to make money/300/jh lee/2023/0
419 hxcj gljh/200/Px Wq/2379/1
420 igwd/200/Fr Qm/2110/1
421 iomc/400/Vc Bo Tt/2348/1
422 jmnz fxwt/100/Tm Mm Ft/2106/1
423 jujv ltnz/300/Xj/2397/0
424 jwhr/300/Lf/1943/1
425 kbag/000/Bk/2325/1
426 kjbw/400/Tx Eg/1906/0
427 kwrq benv/700/Ho Jz/2118/1
428 kyls smdi/300/Fg Pp/2273/1
429 mcdo xuhk/400/Yi/2328/0
430 mcof hrpt/400/Cf Pn/2308/1
431 mwyi qoah/700/Pq/2346/1
432 mzaf/200/Mt Dv/2024/0
433 nguw jhpp/000/Rr Bs Kn/1953/1

```

PRINT_BP 명령어를 입력하니 B+-Tree의 모든 데이터가 책 제목 순으로 출력되었음을 볼 수 있습니다.


```
420  nuIU/100/Yx Ej Yb/2244/1
421  oosy xmsa/200/Iz Yq/2271/1
422  oqzm vnby/500/0s Zf Uu/2399/0
423  ovyj/100/Qa/2303/1
424  peCS oktc/300/Tu Cr/2131/0
425  pkro/100/Qc/2322/0
426  pvtd/300/Xb Jy Kf/2268/0
427  qnvH wrtL/500/Dz Wo/2167/1
428  qspn/400/Vf/2172/1
429  qygv gbtF/400/Bi/2345/1
430  qzmh/700/Uw Cz Uu/2032/1
431  rguq bsmm/000/Uy Xd/2283/1
432  rmmf/600/Nk Gd Wu/2337/1
433  swzu/100/Us Uo Jq/2235/1
434  tbgy/000/Sb/2349/1
435  tkca yvwd/700/Fp Ki Af/2030/1
436  tojy/500/Yx/2185/0
437  tsmr/000/Kz Vs/2016/1
438  ttka lyno/100/Ek Hx Pk/2144/1
439  uqrm/000/Lu Ly/2267/0
440  urzk xqiW/400/Bm/2307/0
441  uvol xrot/100/Kz Iw/2212/0
442  vdvS/000/Sh/2349/0
443  vfph/700/Bu Da Ad/2092/1
444  vlfc qlai/300/Dv Xd/2020/1
445  vylk/300/Vn On Xl/2361/0
446  whap vngg/700/0l Gu/2326/1
447  wwps/500/Mn Cf Xn/2231/1
448  wynq/000/Cs Ex Mg/2104/1
449  xbhM exef/100/Vm Sz Sy/2271/1
```

이어서 PRINT_BP의 결과

```
xhai bynf/600/Sv Im Ld/2220/1
zcjm fiva/200/Su Mh/1926/1
zdvt qnfe/600/Wu Tz EL/2264/0
zjep rucx/500/Up Rn Gz/2032/1
zjhg hmdm/600/Tr Vw/2010/1
zwag kkpw/500/Dq/2374/0
zxil/500/Tb Di Qw/2296/0
=====
```

여기까지 PRINT_BP의 결과

```
95 PRINT_ST 000
96 PRINT_ST 100
97 PRINT_ST 200
98 PRINT_ST 300
99 PRINT_ST 400
```

```
=====PRINT_ST=====
glul/000/Xp Us/2285/3
ixdg khsy/000/Av Te/2291/3
pohy/000/Ip Pp/1977/3
uwab slur/000/Vj Wk Bq/2152/3
ykhk alss/000/Qq Nt Rx/2038/3
=====
```

```
=====PRINT_ST=====
lgaf/100/Dd Bo/2070/3
=====
```

```
=====PRINT_ST=====
fmvn/200/Ho/2010/3
owyt/200/Qd Id Nj/2313/3
ywlx crqh/200/Wb/1986/3
=====
```

```
=====PRINT_ST=====
bvwm/300/Bb Fd Wt/2026/4
cluh iers/300/Ns Ow/2136/4
cnrm yfoq/300/Dd Ml/1909/4
dapx/300/Sj Xa Nr/2241/4
vqdq/300/Oq Ht Lu/2200/4
=====
```

```
=====PRINT_ST=====
eylx/400/Id/1961/4
fuxn tlch/400/Wp Ws/2253/4
lmng/400/Ot Vr/2023/4
nwxv jhna/400/Ze/1971/4
vlaf owxw/400/Nq Mk Gp/2334/4
=====
```

코드가 000~400인 대출 불가 도서를 코드 순서대로 출력한 결과입니다.

```
=====PRINT_ST=====
auby/500/Ii Ww/2201/2
fesz/500/Aq Xg Ea/2058/2
siyg bqew/500/Sx/1919/2
yzxy/500/0c 0h/2042/2
=====

=====PRINT_ST=====
nsbv emku/600/Nq Is/2151/2
qayc vwjg/600/Vp Td Wx/1980/2
=====

=====PRINT_ST=====
awvp/700/Sz/2185/2
fgeg ulnn/700/Ox Xq/2324/2
qjfy/700/Nc Ym Gy/2248/2
twdh zivi/700/Pt/2328/2
wcah/700/Gp Ut/2385/2
=====
```

100	PRINT_ST	500
101	PRINT_ST	600
102	PRINT_ST	700

코드가 500~700인 대출 불가 도서를 책 제목 순으로 출력하였습니다.

	513	=====DELETE=====
	514	Success
	515	=====
	516	
	517	=====DELETE=====
	518	Success
	519	=====
	520	
	521	=====DELETE=====
	522	Success
	523	=====
	524	
	525	=====PRINT_ST=====
	526	glul/000/Xp Us/2285/3
	527	ixdg khsy/000/Av Te/2291/3
	528	pohy/000/Ip Pp/1977/3
	529	uwab slur/000/Vj Wk Bq/2152/3
	530	ykhk alss/000/Qq Nt Rx/2038/3
	531	=====
	532	
	533	=====PRINT_ST=====
	534	lgaf/100/Dd Bo/2070/3
	535	=====
	536	
	537	=====PRINT_ST=====
	538	fmvn/200/Ho/2010/3
	539	owyt/200/Qd Id Nj/2313/3
	540	ywlx crqh/200/Wb/1986/3
	541	=====
	542	
	543	=====PRINT_ST=====
	544	cluh iers/300/Ns Ow/2136/4
	545	cnrm yfoq/300/Dd ML/1909/4
	546	dapx/300/Sj Xa Nr/2241/4
	547	vqdq/300/0q Ht Lu/2200/4
	548	=====
	549	
103	DELETE	
104	DELETE	
105	DELETE	
106	PRINT_ST	000
107	PRINT_ST	100
108	PRINT_ST	200
109	PRINT_ST	300

Selection Tree에서 최상위 우선순위를 가지는 데이터 3개를 지웠고, 다시 000~700의 도서들을 출력한 결과입니다. 모든 힙의 데이터를 알파벳 순으로 나열했을 때 맨 앞 3개의 책은 코드 300의 bvwm, 코드 500의 auby, 코드 700의 awvp입니다. 모든 힙의 데이터를 출력한 결과를 보면 이 3개의 도서 데이터만 지워진 것을 볼 수 있습니다.

```
=====PRINT_ST=====
eylx/400/Id/1961/4
fuxn tlch/400/Wp Ws/2253/4
lmng/400/Ot Vr/2023/4
nwxv jhna/400/Ze/1971/4
vlaf owxw/400/Nq Mk Gp/2334/4
=====

=====PRINT_ST=====
fesx/500/Aq Xg Ea/2058/2
siyg bqew/500/Sx/1919/2
yzxy/500/0c Oh/2042/2
=====

=====PRINT_ST=====
nsbv emku/600/Nq Is/2151/2
qayc vwjg/600/Vp Td Wx/1980/2
=====

=====PRINT_ST=====
fgeg ulnn/700/0x Xq/2324/2
qjfy/700/Nc Ym Gy/2248/2
twdh zivi/700/Pt/2328/2
wcah/700/Gp Ut/2385/2
=====

110 PRINT_ST 400
111 PRINT_ST 500
112 PRINT_ST 600
113 PRINT_ST 700
114 EXIT
=====EXIT=====
Success
=====
```

위 사진과 이어지는 결과입니다. EXIT 명령어를 읽으니 성공적으로 출력이 종료된 것을 볼 수 있습니다.

5. Consideration

- A. 위 내용에 더해 B+-Tree의 삭제를 구현해야 했으나, 구현이 어려워 포기하고 ADD 명령어를 통해 추가 대출된 도서가 대출 가능 권수를 넘으면 B+-Tree에서 데이터만 삭제하고, 이를 Selection Tree로 넘기도록 설계하였습니다. 정상적인 탐색을 위해서는 B+-Tree에서 삭제가 일어날 때 데이터뿐만 아니라 키도 삭제하고, 노드를 병합하는 과정도 구현해야 했으나 데이터만 지우도록 설계한 것이 아쉬움이 남습니

다. 학기가 끝나고 시간이 된다면 B+-Tree의 삭제도 구현하고 싶습니다.