**Data Structures**

**Spring 2020**

**Assignment #2**

**Date: 2020. 06. 24**

**ID/Name: 201533661 이승수**

**[6-sorting p.50,51] (Quick Sort)**

**(1-1) Exercise (show all the steps)**

**sort the list: (pivot=last element)(\* do not split up the P list already formed \*)**

**85, 17, 63, 45, 17, 31, 85, 50**

**[Answer]**

|  |  |
| --- | --- |
| Pass1 | 85, 17, 63, 45, 17, 31, 85, (50)  85, 17, 63, 45, 17, 31, 85, (50)  31, 17, 63, 45, 17, 85, 85, (50) (L/R cursor exchanged)  31, 17, 63, 45, 17, 85, 85, (50)  31, 17, 17, 45, 63, 85, 85, (50) (L/R cursor exchanged)  31, 17, 17, 45, 63, 85, 85, (50)  L/R cursor met, so R cursor stop at 45, and exchange pivot(50)  with L cursor at 63.  31, 17, 17, 45, (50), 85, 85, 63 |
| Pass2 | At rest of passes, sort one by one at each left/right side of pivot(50).  31, 17, 17, (45, 50, 63), 85, 85 |
| Pass3 | 17, 17, (31, 45, 50, 63, 85), 85 |
| Pass4 | 17, (17, 31, 45, 50, 63, 85, 85) |

Result: 17, 17, 31, 45, 50, 63, 85, 85

**(1-2)Exercise (show all the steps)**

**The first element of the list may be selected as pivot to do descending order sorting.**

**In this case, the description on page 45 (the L cursor and R cursor) is reversed in direction. \*\* This is left as exercise.**

**sort the list: (pivot=first element)**

**85, 24, 63, 45, 17, 31, 96, 50**

**[Answer]**

|  |  |
| --- | --- |
| Pass1 | (85), 24, 63, 45, 17, 31, 96, 50  (85), 96, 63, 45, 17, 31, 24, 50 (L/R cursor exchanged)  (85), 96, 63, 45, 17, 31, 24, 50  L/R cursor met at 63, so L cursor stopped, exchange pivot(85) with R cursor at 96.  96, (85), 63, 45, 17, 31, 24, 50 |
| Pass2 | (96, 85, 63), 45, 17, 31, 24, 50 |
| Pass3 | (96, 85, 63, 50), 17, 31, 24, 45 |
| Pass4 | (96, 85, 63, 50, 45), 31, 24, 17 |
| Pass5 | (96, 85, 63, 50, 45, 31), 24, 17 |
| Pass6 | (96, 85, 63, 50, 45, 31, 24), 17 |

Result: 96, 85, 63, 50, 45, 31, 24, 17

<Heap Sort>

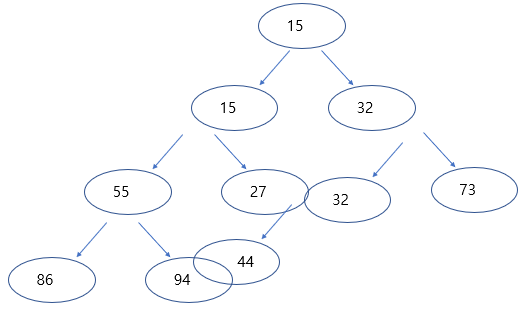
(2) **Exercise 1**

(1) Create a min heap with the following keys

15 86 32 55 27 32 73 15 94 44

**[Answer]**

* [15 15 32 55 27 32 73 86 94 44]



(2) Do a heap sort (in ascending order)

**[Answer]**

Pop: 15 -> heap: [15 27 32 55 44 32 73 86 94]

Pop: 15 -> heap: [27 44 32 55 94 32 73 86]

Pop: 27 -> heap: [32 44 32 55 94 88 73]

Pop: 32 -> heap: [32 44 73 55 94 88]

Pop: 32 -> heap: [44 55 73 88 94]

Pop: 44 -> heap: [55 88 73 94]

Pop: 55 -> heap: [73 88 94]

Pop: 73 -> heap: [88 94]

Pop: 88 -> heap: [94]

Pop: 94 -> heap: []

Heap sort Result: [15 15 27 32 32 44 55 73 88 94]

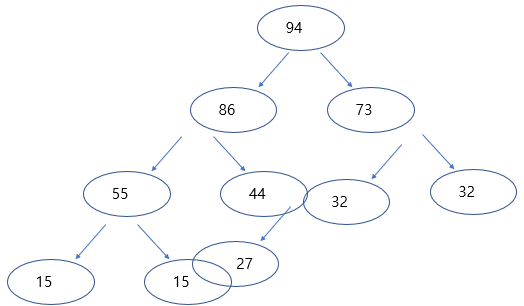
**Exercise 2**

(1) Create a max heap with the following keys

15 86 32 55 27 32 73 15 94 44

**[Answer]**

-> [94 86 73 55 44 32 32 15 15 27]



(2) Do a heap sort (in descending order)

**[Answer]**

Pop: 94 -> heap: [86 55 73 27 44 32 32 15 15]

Pop: 86 -> heap: [73 55 32 27 44 32 15 15]

Pop: 73 -> heap: [55 44 32 27 15 32 15]

Pop: 55 -> heap: [44 27 32 15 15 32]

Pop: 44 -> heap: [32 27 32 15 15]

Pop: 32 -> heap: [32 27 15 15]

Pop: 32 -> heap: [27 15 15]

Pop: 27 -> heap: [15 15]

Pop: 15 -> heap: [15]

Pop: 15 -> heap: []

Heap sort Result: [94 73 55 44 32 32 27 15 15]

**Exercise (Radix Sort) -lec8: p.46,47,50**

**LSD Radix Sort**

**sort the list: 170000, 10**

**[Answer]**

Pass1: 170000, 10

Pass2: 170000, 10

Pass3,4,5: [10], 170000

Pass6: [10, 170000]

Radix Sort Result: [10, 170000]

**sort the list: 17000, 190, 2, 3802, 24, 45, 75, 66**

**[Answer]**

Pass1: 17000, 190, 2, 3802, 24, 45, 75, 66

Pass2: [2], 17000, 3802, 24, 45, 66, 75, 190

Pass3: [2, 24, 45, 66, 75], 17000, 190, 3802

Pass4: [2, 24, 45, 66, 75, 190], 3802, 17000

Pass5: [2, 24, 45, 66, 75, 190, 3802], 17000

Pass6: [2, 24, 45, 66, 75, 190, 3802, 17000]

Radix Sort Result: [2, 24, 45, 66, 75, 190, 3802, 17000]

**MSD Radix Sort: Exercise**

**sort the list: 170, 4, 160, 90, 4, 45, 24, 890, 892, 802, 2, 45**

**[Answer]**

Start: 170, 004 160, 090, 004, 015, 024, 890, 892, 802, 002, 045

Pass1: [004, 090, 004, 015, 024, 002, 045], [170, 160], [890, 892, 802]

Pass2: [004, 004, 002], [015], [022], [045], [090], [160], [170], [802], [890, 892]

Pass3: [002] [004, 004], [015], [022], [045], [090], [160], [170], [802], [890], [892]

MSD Radix sort Result: [002, 004, 004, 015, 022, 045, 090, 160, 170, 802, 890, 892]

**[WHW5 (lec8: p.63~66)] (20 points)**

**WHW-5(h1): (5 points)**

**Descending Order Heap Sort (Max Heap)**

**31, 11, 3, 20, 19, 24, 17, 3, 31**

**[Answer]**

Start: [31 31 24 20 19 13 17 3 11]

Pop:31 -> heap: [31 20 24 11 19 13 17 3]

Pop:31 -> heap: [24 20 17 11 19 13 3]

Pop:24 -> heap: [20 19 17 11 3 13]

Pop:20 -> heap: [19 13 17 11 3]

Pop:19 -> heap: [17 13 3 11]

Pop:17 -> heap: [13 11 3]

Pop:13 -> heap: [11 3]

Pop:11 -> heap: [3]

Pop:3 -> heap: []

Max heap sort Result: [31 31 24 20 19 17 13 11 3]

**WHW-5(h2): (5 points)**

**Ascending Order Heap Sort (Min Heap)**

**31, 11, 3, 20, 19, 24, 17, 3, 31**

**[Answer]**

Start: [3 3 17 11 19 24 31 20 31]

Pop:3 -> heap: [3 11 17 20 19 24 31 31]

Pop:3 -> heap: [11 19 17 20 31 24 31]

Pop:11 -> heap: [17 19 24 20 31 31]

Pop:17 -> heap: [19 20 24 31 31]

Pop:19 -> heap: [20 31 24 31]

Pop:20 -> heap: [24 31 31]

Pop:24 -> heap: [31 31]

Pop:31 -> heap: [31]

Pop:31 ->heap: []

Min heap sort Result: [3 3 11 17 19 20 24 31 31]

**WHW-5(h3): (5 points)**

**LSD Radix Sort**

**310, 11, 3, 20, 11119, 24, 17, 3, 31**

**[Answer]**

Pass1: 310, 20, 11, 31, 3, 3, 24, 17, 11119

Pass2: [3, 3], 310, 11, 17, 11119, 20, 24, 31

Pass3: [3, 3, 11, 17, 20, 24, 31], 310, 11119

Pass4: [3, 3, 11, 17, 20, 24, 31, 310], 11119

Pass5: [3, 3, 11, 17, 20, 24, 31, 310, 11119]

LSD Radix Sort Result: [3, 3, 11, 17, 20, 24, 31, 310, 11119]

**WHW-5(h4): (5 points)**

**MSD Radix Sort**

**310, 11, 3, 20, 11119, 24, 17, 3, 31**

**[Answer]**

Start: 00310, 00011, 00003, 00020, 11119, 00024, 00017, 00003, 00031

Pass1,2: [00310, 00011, 00003, 00020, 00024, 00017, 00003, 00031], [11119]

Pass3: [00011, 00003, 00020, 00024, 00017, 00003, 00031], [00310], [11119]

Pass4: [00003, 00003], [00011, 00017], [00020], [00031], [00310], [11119]

Pass5: [00003, 00003], [00011], [00017], [00020], [00031], [00310], [11119]

MSD Radix Sort Result: [3, 3, 11, 17, 20, 31, 310, 11119]