

Bucket/radix sorting

5/18/26

Administrivia

- HW 7 (fork/join and counting query frequency) due tomorrow night

Recall/redo: Implementing heap sort

- Can turn array into heap by running downheap on each element, starting at end (“heapifying” an array)
- As values are removed from the heap, it shrinks and frees up space in the array

Final note on PQs

- Can also use a balanced binary search tree (like TreeSet), with key being the priority
- Allows in-order iteration in linear time

Bucket sort

- Use an array to store the number of elements that have each value (count of value x goes into cell x)
- Traverse the array, printing an appropriate number of copies of each value

Radix sort

- Perform bucket sort one digit at a time
- Counter-intuitive part:

Do this least significant digit first

Example: Radix sort the numbers below

Orig.

42

13

2

20

21

14

24

30

Example: Radix sort the numbers below

Orig.

0

1

2

3

4

42

13

2

20

21

14

24

30

Example: Radix sort the numbers below

Orig.

0

1

2

3

4

42

13

2

20

21

14

24

30

Which column does the 42 go into?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

Example: Radix sort the numbers below

Orig.

0

1

2

3

4

42

13

2

20

21

14

24

30

Which column does the 42 go into?

A. 0

B. 1

C. 2

D. 3

E. 4

Example: Radix sort the numbers below

<u>Orig.</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42			42		
13					
2					
20					
21					
14					
24					
30					

Example: Radix sort the numbers below

<u>Orig.</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42			42	13	
13					
2					
20					
21					
14					
24					
30					

Example: Radix sort the numbers below

<u>Orig.</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42			42	13	
13			2		
2					
20					
21					
14					
24					
30					

Example: Radix sort the numbers below

<u>Orig.</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20	21	42	13	14
13	30		2		24
2					
20					
21					
14					
24					
30					

Example: Radix sort the numbers below

<u>Orig.</u>	<u>Pass 1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20	20	21	42	13	14
13	30	30		2		24
2	21					
20	42					
21	2					
14	13					
24	14					
30	24					



Copy the columns to get result of first pass

Example: Radix sort the numbers below

<u>Orig.</u>	<u>Pass 1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20					
13	30					
2	21					
20	42					
21	2					
14	13					
24	14					
30	24					

Example: Radix sort the numbers below

<u>Orig.</u>	<u>Pass 1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20			20		
13	30					
2	21					
20	42					
21	2					
14	13					
24	14					
30	24					

Example: Radix sort the numbers below

<u>Orig.</u>	<u>Pass 1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20			20	30	
13	30					
2	21					
20	42					
21	2					
14	13					
24	14					
30	24					

Example: Radix sort the numbers below

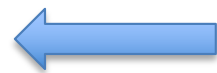
<u>Orig.</u>	<u>Pass 1</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20	2	13	20	30	42
13	30		14	21		
2	21			24		
20	42					
21	2					
14	13					
24	14					
30	24					



Note that 2 has 0 for its tens digit!

Example: Radix sort the numbers below

<u>Orig.</u>	<u>Pass 1</u>	<u>Final</u>	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
42	20	2	2	13	20	30	42
13	30	13		14	21		
2	21	14			24		
20	42	20					
21	2	21					
14	13	24					
24	14	30					
30	24	42					



Copy the columns to get final result

What ADTs would you use for this problem?

- A. List or Priority Queue
- B. Queue or Stack
- C. Map
- D. Set
- E. More than one of the above