CSCI-GA.1170-001/002 Fundamental Algorithms October 14, 2015

Solutions to Problem 4 of Homework 5 (9 points)

Name: GOWTHAM GOLI (N17656180) *Due: Tuesday, October 13*

For each example choose one of the following sorting algorithms and carefully justify your choice: HEAPSORT, RADIXSORT, COUNTINGSORT. Give the expected runtime for your choice as precisely as possible. If you choose Radix Sort then give a concrete choice for the basis (i.e. the value of "r" in the book) and justify it. (**Hint**: We assume that the array itself is stored in memory, so before choosing the fastest algorithm, make sure you have the space to run it!)

(a) Sort the length 2^{16} array A of 128-bit integers on a device with 100MB of RAM.

Solution:

Given, $n = 2^{16}$

Memory needed to store array of length 2^{16} of 128-bit integers $= 2^{23}$
RAM capacity is 100 MB $= 100 \times 2^{10}$ KB $= 100 \times 2^{20}$ B $= 100 \times 2^{23}$ bits

Time taken to sort the given array using heap sort $= O(n \log n) = O(2^{16}.16) = O(2^{20})$
Extra space required using heap sort $= O(1)$

$b = 128 > 16 = \log n \therefore$ choose $r = \log_n = 16$
Time taken to sort the given array using radix sort $= O(2(bn)/(\log n)) = O(256 \times 2^{16}/16) = O(2^{20})$
Extra space required using radix sort $= O(n + 2^r) = O(2^{16} + 2^{16}) = O(2^{17}))$

Time taken to sort the given array using counting sort $= O(n + k) = O(2^{16} + 2^{128})$
Extra space required using counting sort $= O(n + k) = O(2^{16} + 2^{128})$

The running time of radix sort and heap sort is the lowest of the three and also there is sufficient space in RAM to allocate the extra space used by radix sort and heap sort. Hence it would be ideal to use heap sort or radix in this case

□

(b) Sort the length 2^{24} array A of 256-bit integers on a device with 600MB of RAM.

Solution:

Given, $n = 2^{16}$

Memory needed to store array of length 2^{24} of 256-bit integers $= 2^{32}$ bits $= 512 \times 2^{23}$ bits
RAM capacity is 600 MB $= 600 \times 2^{10}$ KB $= 600 \times 2^{20}$ B $= 600 \times 2^{23}$ bits

Time taken to sort the given array using heap sort $= O(n \log n) = O(2^{24}.24) = O(24 \times 2^{24})$
Extra space required using heap sort $= O(1)$

$b = 256 > 24 = \log n \therefore$ choose $r = \log_n = 24$
Time taken to sort the given array using radix sort $= O(2(bn)/(\log n)) = O(512 \times 2^{24}/24) =$

$O(64 \times 2^{24}/3)$
Extra space required using radix sort $= O(n + 2^r) = O(2^{24} + 2^{24}) = O(2^{25}))$

Time taken to sort the given array using counting sort $= O(n + k) = O(2^{24} + 2^{256})$
Extra space required using counting sort $= O(n + k) = O(2^{24} + 2^{256})$

It is quite clear that there is no sufficient memory in RAM to allocate the extra space required for radix sort and counting sort. So the only option here is to use heap sort

$\square$

(c) Sort the length 2^{16} array A of 16-bit integers on a device with 1GB of RAM.

Solution:

Given, $n = 2^{16}$

Memory needed to store array of length 2^{16} of 16-bit integers $= 2^{20}$ bits
RAM capacity is 1 GB $= 2^{10}$ MB $= 2^{20}$ KB $= 2^{30}$ B $= 2^{33}$ bits

Time taken to sort the given array using heap sort $= O(n \log n) = O(2^{16}.2^4) = O(2^{20})$
Extra space required using heap sort $= O(1)$

$b = 16 = 16 = \log n \therefore$ choose $r = \log_n = 16$ Time taken to sort the given array using radix sort $= O(2(bn)/(\log n) = O(32 \times 2^{16}/16) = O(2^{17})$
Extra space required using radix sort $= O(n + 2^r) = O(2^{16} + 2^{16}) = O(2^{17}))$

Time taken to sort the given array using counting sort $= O(n + k) = O(2^{16} + 2^{16}) = O(2^{17})$
Extra space required using counting sort $= O(n + k) = O(2^{16} + 2^{16}) = O(2^{17})$

counting sort and radix sort takes the least time to sort the given array and also there is sufficient space in the RAM to allocate the extra space that counting sort and radix sort takes. Hence it would be ideal to use counting sort or radix sort in this case(In this case, radix sort = counting sort)

$\square$