(1)

N=1 passes 1 1,2,3,3,4,4,4,4,5,5,5,5,

N=2 passe 2 5,5,5,5

N=3 passes 3

N=4 passes 3 # passes = Ceil(log2(n+1)) + 1

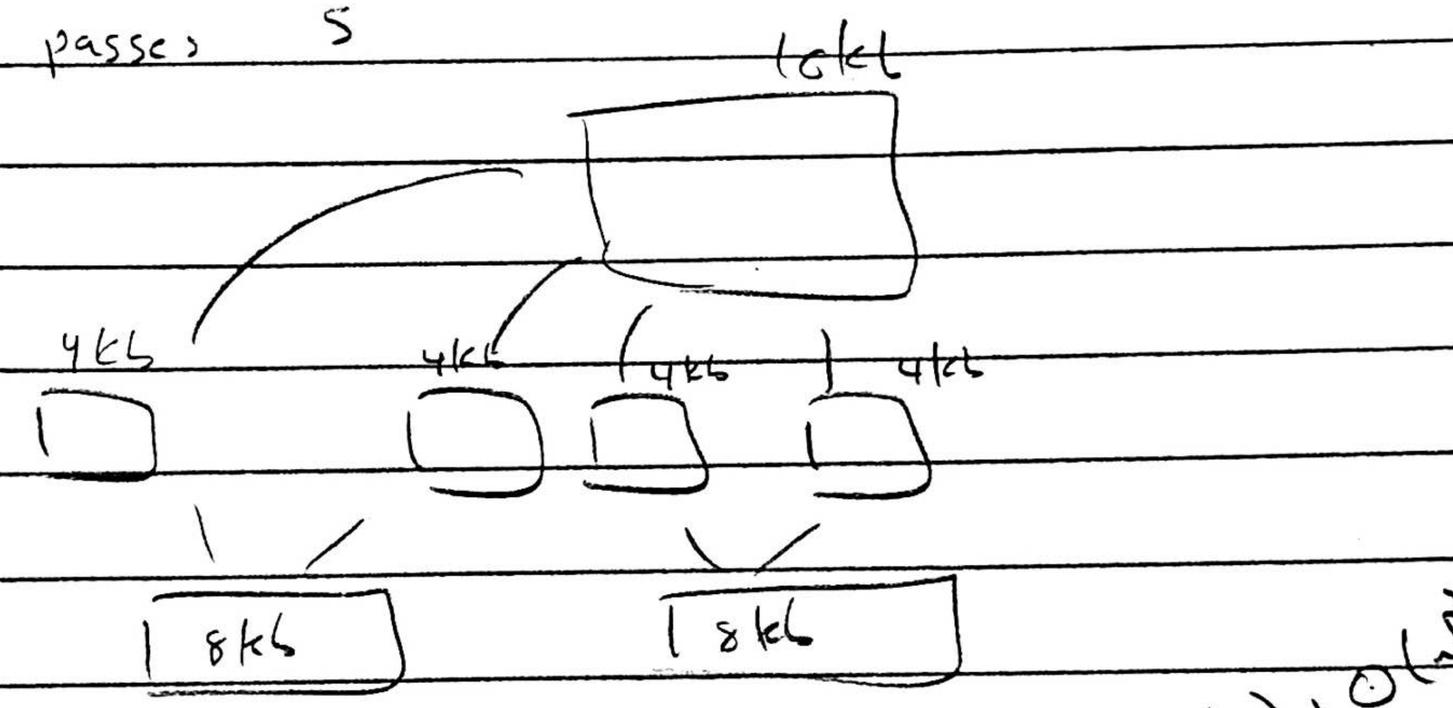
N=5 passes 4 or

N=6 passes 4 a[n] = floor(n/2)) + 1

N=7 passes 4 a(n) = a(n-1)

N=8 Passes 4

N=9 passes 5



$T(n) = AT(n/B) + O(n^D)$

Y master theorem

substitution

$2T\left(\frac{n}{4}\right) + \log(n) + T\left(\frac{2(n+2)}{2}\right)$

$\Rightarrow t(n) = 7t\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right)$

$\Rightarrow 8t(0) + \frac{17}{6}n\,t'(0) + \frac{32}{72}n^2 t''(0) + \frac{83n^3 t'''(0)}{1296} + \frac{193n^4 t^4(0)}{31104} + \cdots$

A: 7

B: 2

D: 2

$n \log n$

$O(n^5)$

(2)

Pseudo Code Quicksort - O(nlogn)

create datastructure of choice. ε

1) fstream file open file

file.open("filename", ios::in)

- while (file.good()){ while file has lines

# of passes depends

Sort everything and place into a data structure

sort() : quicksort

everything happens here

f.close

f.open(name, output)

open file for output

for(auto x: "data structure"){

file << x << endl;

}

(3)

datastructure +1

fstream file +1

open file +1

read file (n)

buffer and sort (n log n)

close file +1

open file to write +1

write to file (n)

close file +1

$T(n) = 2n + n\log(n) + 6$ ← Time Complexity

$O(n \log n)$ ← Space Complexity

theorem

$t(n) = 2n + n\log n + 6$

$t(n) = a\,t(\frac{n}{s}) + f(n)$

$A = 2$

$B = 1$

$D = 1$

$n \log n$