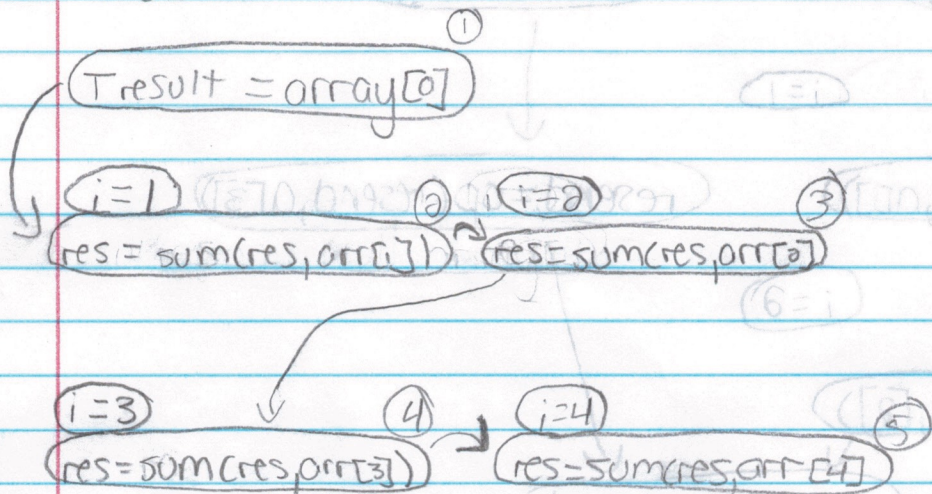


Reduce

Roman Styrku

3.1a size + n = 5



Width = 1

Critical Path = n

Work = n

3.1b T total = 0

3.1c Tresult = arr[0];

Tresultend = arr[n-1];

size = n/P;

for (int i=0; i < arr.length; i += size) {

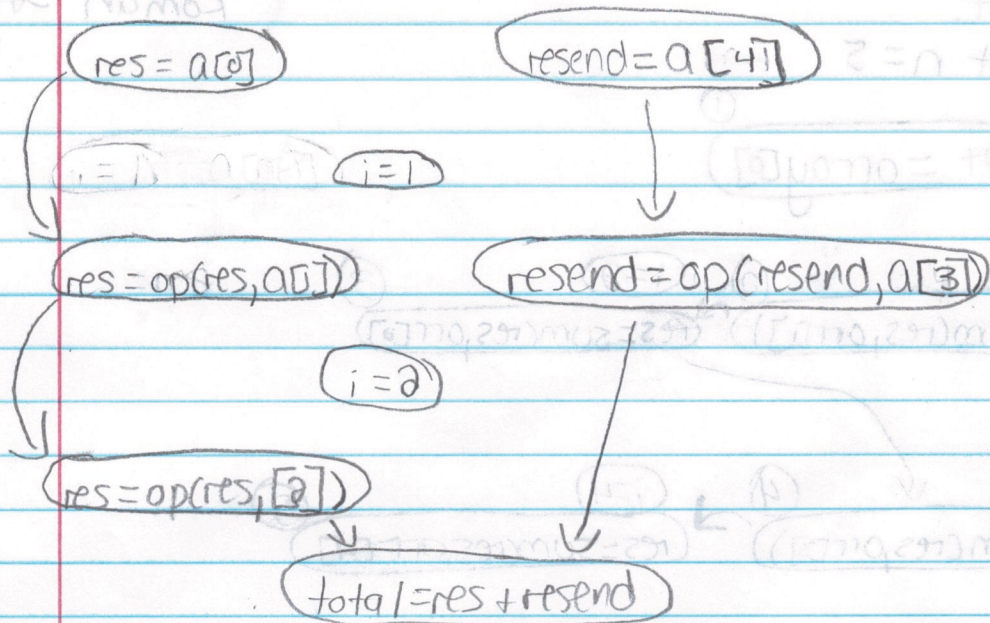
for (int j=0; j < size; j++) {

Ttotal += arr[j];

}

}

return Ttotal;



Width = P

Work = $n + P$

CP = $n/P + 1$

3.10

P1	res = a[0]	res = op(res, a[1])	res = op(res, a[2])
P2	resend = a[4]	resend = op(resend, a[3])	
	1	2	

On P Processors it would divide it by P sections and work its way down to the total.

Reduce Sheet 2

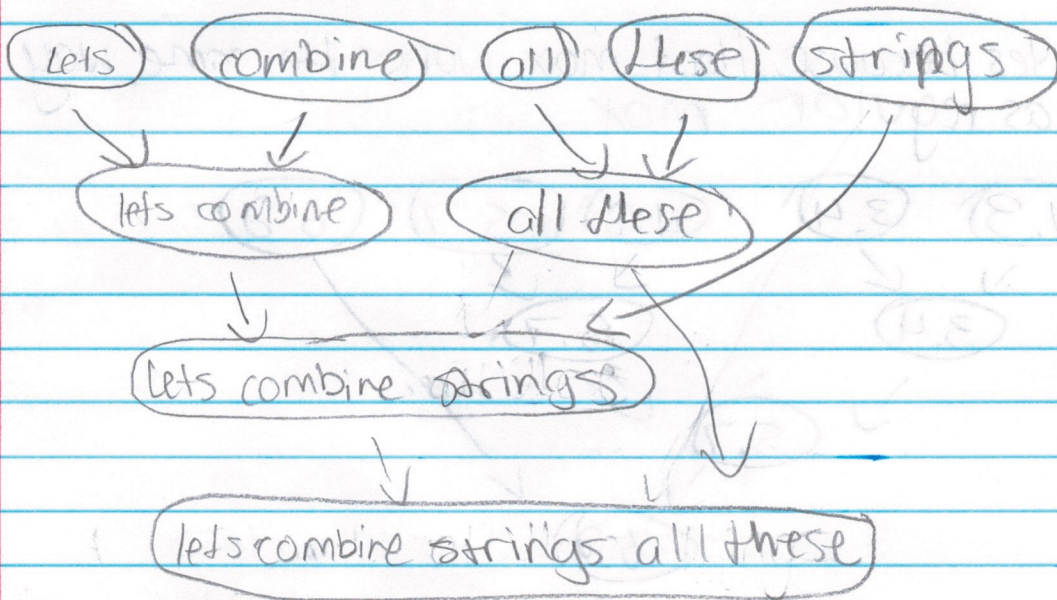
3.2a. Yes because max can also be split up and combined

$\max(a[0], a[1]) \quad \max(a[2], a[3]) \quad \max(a[4], a[5])$

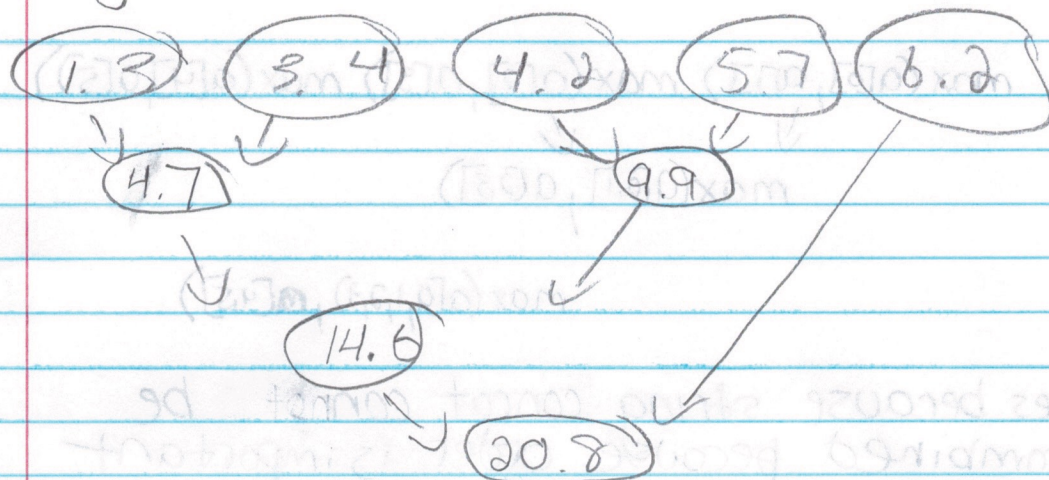
$\downarrow \quad \downarrow$
 $\max(a[0,1], a[2,3])$

$\max(a[0,1,2,3], a[4,5])$

3.2b. yes because string concat cannot be combined because order is important



3.2 c. Yes because float addition works the same way.



3.2 d Yes because float max works the same way as regular max

