

ID	Result	Vote	Exercise	Percentage	Points	Passed tests?	Notes
196488	NOT PASSED	1	A1.1 rank_categories	20.00%	0.8 / 4		+ some ideas
			A1.2 plot	0.00%	0 / 2		- much confusion
			A1.3 enrich	0.00%	0 / 3		some code but no plot
			A2 dump	0.00%	0 / 6	n	no code
			B1 theory	0.00%	0 / 5		empty
			B2.1 bubble_sort	0.00%	0 / 5	n	no code
			B2.2 merge	0.00%	0 / 5	n	no code
			B3.1 stacktris_shorten	0.00%	0 / 5	n	no code
			B3.2 stacktris drop1	0.00%	0 / 7	n	no code
			B3.2 stacktris drop2h	0.00%	0 / 8	n	no code
			style bonus	0.00%	0 / 1		
c.g.	NOT PASSED	13	A1.1 rank_categories	100.00%	4 / 4		
			A1.2 plot	100.00%	2 / 2		
			A1.3 enrich	0.00%	0 / 3		- very little code
							Found two versions, both wrong
							For the first one:
							- does not decrease the quantity in the truck at each iteration
							- useless distinction between j=0 and j>0
							- does not return early when there is no remaining waste to unload
			A2 dump	50.00%	3 / 6	n	O(n^3)
			B1 theory	50.00%	2.5 / 5		- there is only one while, there should have been two (it's even written in exercise text)
							- you are swapping the *pointers*, instead, you should have swapped *data* inside nodes
B2.1 bubble_sort	0.00%	0 / 5	n	- it's just appending full lists instead of interleaving elements			
				- calling bubble_sort which is O(n^2), it can't possibly complete in linear time as required by this exercise			
B2.2 merge	20.00%	1 / 5	n				
B3.1 stacktris_shorten	0.00%	0 / 5	n	no code			
B3.2 stacktris drop1	0.00%	0 / 7	n	no code			
B3.2 stacktris drop2h	0.00%	0 / 8	n	no code			
style bonus	0.00%	0 / 1					

205223 NOT PASSED

11

A1.1 rank_categories	10.00%	0.4 / 4
A1.2 plot	100.00%	2 / 2
A1.3 enrich	20.00%	0.6 / 3
A2 dump	60.00%	3.6 / 6
B1 theory	50.00%	2.5 / 5
B2.1 bubble_sort	20.00%	1 / 5
B2.2 merge	0.00%	0 / 5
B3.1 stacktris _shorten	20.00%	1 / 5
B3.2 stacktris drop1	0.00%	0 / 7
B3.2 stacktris drop2h	0.00%	0 / 8
style bonus	0.00%	0 / 1

- builds huge string each round
- wrong filtering of stopwords
- missing histogram counting
- missing list sorting

- modifying old db
- using lowercase instead of uppercase
- if  $dp \leq q$  ----> if  $dp < q$
- always completely fills the cell instead of checking current available amount
- doesn't check if there is still waste not disposed
- $O(n^3)$
- using a single pointer for both whiles
- swapping nodes instead of the data
- overcomplex
- introduced \_last pointer which is not updated in .add method
- runs in circa  $O(n1*n2*\max(n1,n2))$  instead of  $O(n1+n2)$
- modifies both input lists, should have only read from them
- scan bottom-to-top, instead of top-down
- wrong full row checking, doesn't consider cases with numbers > 1 like [022]
- forgetting to pop
- adding a list to ret, which becomes a list of lists instead of a single list
- missing opportunity of early return, which is faster

no code  
no code