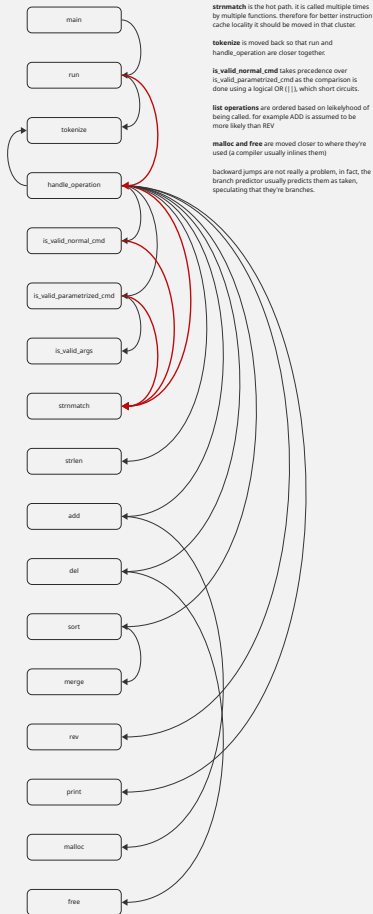## NAIVE APPROACH

CACHE FRIENDLY

main

run

tokenize

handle_operation

is_valid_normal_cmd

is_valid_parametrized_cmd

is_valid_args

strnmatch

strlen

add

del

sort

merge

rev

print

malloc

free

strnmatch is the hot path. it is called multiple times by multiple functions. therefore for better instruction cache locality it should be moved in that cluster.

tokenize is moved back so that run and handle_operation are closer together.

is_valid_normal_cmd takes precedence over is_valid_parametrized_cmd as the comparison is done using a logical OR (| |), which short circuits.

list operations are ordered based on leikelyhood of being called. for example ADD is assumed to be more likely than REV

malloc and free are moved closer to where they're used (a compiler usually inlines them)

backward jumps are not really a problem, in fact, the branch predictor usually predicts them as taken, speculating that they're branches.

main

tokenize

run

handle_operation

strnmatch

is_valid_normal_cmd

is_valid_parametrized_cmd

is_valid_args

strlen

add

malloc

del

free

print

sort

merge

rev