

### Part E: Programming Techniques

One programming technique that we both would like to continue incorporating into our programming practice is frequently using valgrind to check for memory leaks. For our first program in this class, Brightness, we failed to valgrind at all since the program was working successfully and we did not malloc anything ourselves. However, in testing, we lost all points for the program since we forgot that Pnmrdr also needs to be freed. Instead of rushing on to the next task after supposedly finishing the program, we should have taken a moment to run valgrind and triple check the Pnmrdr interface rules. Even when writing the simlines program, we did not valgrind until the program was completed, which led to 2 hours of debugging the memory leaks. For this assignment, we ran valgrind after writing each function and caught memory problems before they became major issues. In the end, this saved us a lot of time and stress chasing down hidden mallocs. Even though for smaller programs like these, the memory used is freed after runtime anyways, it is extremely important for us as young programmers to get in the habit of cleaning up our own memory so that when we are programming a piece of a much larger project, missed memory leaks don't impact the performance of the overall system.

For those who do not have experience with valgrind, it is a built in tool that keeps track of memory that is allocated and freed by your program. Not freeing used memory means that there is less memory available for other programs that need to run and can cause problems down the line. To use valgrind from the terminal, simply type "valgrind" before the typical run command for the program (./program\_name or ./a.out). You can also run it with the tag --leak-check=full, and the resulting output will show you exactly where the program allocated memory that was not freed. The most serious memory leaks will be shown in the "definitely lost" category. This is memory that you completely lost on the heap and is the most severe kind. "Still reachable" shows memory that you still have the potential to free, but did not actually free in the program. Watch for valgrind errors, which are even more fatal than memory leaks and should be fixed immediately. The leak-check full tag will point to where those errors, if any, are occurring.