E0 251: Programming Assignment 5
Due before 11.59 PM on 25-11-2019

Please mail the source code with a README file and test cases (code and test data in separate files) as well as a brief PDF report (see (f) below) to ullas@iisc.ac.in and protikpaul@iisc.ac.in. Please make a ZIP file of all the files and then mail the ZIP file. Any extra instructions regarding submission (if necessary) will be sent by Ullas and Protik.

[Shen, Peterson, `A Weighted Buddy Method for Dynamic Storage Allocation', CACM, October 1974, pgs 558-562] proposes a variation on the exponential buddy system that uses fixed block sizes of 2^k and 3×2^k. Splitting a block results in 2 blocks of different sizes. In this assignment, you will implement Shen and Peterson's scheme, which we will call the SPheap allocator.

a) [SPheap] Write a C function to initialize the memory area and data structures to be used by the SPheap allocator, viz, SPheap memory space, Available Space List, as well as KVAL, TAG and TYPE fields for its blocks. The KVAL, TAG and TYPE fields will not be stored within the SPheap blocks but centrally. Do the dynamic memory allocations for these using standard library malloc() calls. Let the size of SPheap allocatable memory be 256MBytes.
b) [SPheap instrumentation] Based on Shen and Peterson's paper, write C functions for allocation and deallocation of space from the SPheap memory space, to be used as replacements for malloc() and free() by programs. Include instumentation to gather the following
    1. Numbers of allocation requests, deallocation requests, SPheap area splits, buddy recombinations
    2. Internal fragmentation, quantified by the (sum of SPheap area allocated for all successful allocation requests minus sum of the request sizes of all successful allocation requests) divided by sum of request sizes of all successful allocation requests
    3. External fragmentation: Abort the execution of a test program when an allocation request cannot be satisfied, with an error message that mentions the 2-tuple (Asize, AllocArea%), where Asize is the size of the allocation request that could not be satisfied, AllocArea% is the total currently allocated SPheap area as a percentage of total SPheap area
c) [OneBin] Write initialization, allocation and deallocation functions in C to implement another dynamic memory allocator called OneBin, a bin-based heap memory allocator which supports only one fixed allocation block size. The initialization function takes one parameter, the fixed allocation block size. Let the size of OneBin allocatable memory be 256MBytes. You must specialize OneBin for the needs of the Assignment 2 (Polynomial Arithmetic) program.
d) [Synthetic request generation] Write a SPheap test program that generates synthetic memory allocation and deallocation requests using the scheme described in Section 3 of Shen and Peterson's paper. Use it to test and study your implementation of SPheap.
e) [Comparison] Test your SPheap and OneBin functions on your polynomial arithmetic program, replacing calls to malloc() and free() by the corresponding calls to SPheap or OneBin functions. Compare the SPheap and OneBin versions in terms of memory space efficiency and speed. Also compare their performance with that of your original standard library malloc/free version of the program.
f) [Report] Write a short (pdf) report on your findings in (d) and (e).

COPYING EITHER FROM INTERNET OR FROM A FRIEND IS A CRIME AND WILL BE DEALT WITH SEVERELY. BOTH PARTIES INVOLVED IN COPYING WILL BE PUNISHED. IISc PROVIDES SOFTWARE TOOLS TO CHECK COPYING.