

CSE 2320 Lab Assignment 2

Due July 19, 2017

Goal:

1. Understanding of dynamic programming.
2. Understanding of subset sums.

Requirements:

1. Design, code, and test a C program that uses *dynamic programming* to partition (if possible) a sequence of n positive integers into three subsequences such that the sum of each subsequence is the same. For example, if the input were (10, 20, 30, 40, 40, 50, 80), with a total of $m = 270$, the three $m/3 = 90$ subsequences could be (10, 80), (20, 30, 40), and (40, 50). If the input were (20, 20, 30, 50), then no solution is possible even though the values yield a sum ($m = 120$) divisible by 3 ($m/3 = 40$).

The input should be read from standard input (which will be one of 1. keyboard typing, 2. a shell redirect (<) from a file, or 3. cut-and-paste. Do NOT prompt for a file name!). The first line of the input is n , the length of the sequence. Each of the remaining lines will include one sequence value.

Your program should echo the input sequence in all cases. The dynamic programming table should be output when $m/3 \leq 10$, but in no other cases. Error messages should be displayed if m is not divisible by 3 or if the problem instance does not have a solution. When a solution exists, it should be displayed with each subsequence in a separate column:

i	0	1	2
1	10		
2			20
3			30
4			40
5		40	
6		50	
7	80		

2. Submit your C program on Blackboard by 12:45 p.m. on Wednesday, July 19. One of the comment lines should include the compilation command used on OMEGA. Another comment should indicate the asymptotic worst-case time in terms of m and n .

Getting Started:

1. If you wanted two sequences summing to $m/2$, then the backtrace part of `subsetSum.c` could *easily* be modified. By finding one subsequence that sums to $m/2$, the remaining elements would be another subsequence that sums to $m/2$. Similarly, your program should use dynamic programming to find *two* subsequences that each sum to $m/3$ and then take the leftover values as the third subsequence. Thus, this is a *two-dimensional* DP situation, not *one-dimensional* like ordinary subset sums in Notes 7.F.
2. Dynamic programming is the only acceptable method for doing this lab.