# TCSS 343 - Week 5

Jake McKenzie

August 21, 2018

**Dynamic Programming**

"Perhaps thinking should be measured not by what you do
but by how you do it."
. . .
Richard Hamming

"For the last sixty five years(speaking in 2018), due to Moore's law, with a
clockwork precision, computer capability has been doubling every year and
a half. Without fast algorithms you cannot bring to bare Moore's Law. A
dramatic increase in computer speed needs to be coupled with efficient
algorithms."
. . .
Christos Papadimitriou

"The best programs are the ones written when the programmer is supposed
to be working on something else."
. . .
Melinda Varian

| i | $w_i$ | $h_i$ | $n_i$ |
|---|---|---|---|
| 1 | 1 | 1 | 3 |
| 2 | 2 | 4 | 2 |
| 3 | 3 | 6 | 2 |
| 4 | 4 | 5 | 1 |
| 5 | 5 | 7 | 1 |
| 6 | 6 | 8 | 1 |



1. Suppose Santa has 6 kinds of toys, each kind of toy has its own weight $w_i$ in tons, happiness rating $h_i$ in ... joy, and quantity $n_i$. Santa would like to maximize the total hapiness of the children but the total weight of his bag cannot exceed 17 tons. Their weight, hapiness rating and quantity are defined above. Please help Santa by filling in the DP table below, where dp[i][j] indicates the maximum value you can get with weight less or equal to $j$ using toys 1 to $i$. What is the final solution to this problem and briefly explain how you came to this solution. To help you get started, 23 was generated by solving the equation $i_1 + 2i_2 + 3i_3 \leqslant 15$ which gives you the most value. That value was found by $3 \cdot 1 + 2 \cdot 4 + 2 \cdot 6 = 23$. 12 was found by solving the equation $i_1 + 2i_2 + 3i_3 + 4i_4 + 5i_5 + 6i_6 \leqslant 6$ which gives you the most value. That value was found by $2 \cdot 6 = 12$

| i\w | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| {1} | 0 | | | | | | | | | | | | | | | | | |
| {1,2} | 0 | | | | | | | | | | | | | | | | | |
| {1,2,3} | 0 | | | | | | | | | | | | | | | 23 | | |
| {1,2,3,4} | 0 | | | | | | | | | | | | | | | | | |
| {1,2,3,4,5} | 0 | | | | | | | | | | | | | | | | | |
| {1,2,3,4,5,6} | 0 | | | | | | 12 | | | | | | | | | | | |

2. In mathematics, a sequence of positive real numbers $s_1$, $s_2$,... is called *superincreasing* if each element in the sequence is greater than the sum of all previous elements in the sequence:

$$s_{n+1} > \sum_{i=1}^{n} s_i$$

For example: $\{2, 3, 7, 16, 65, 321, 4546\}$ is a superincreasing sequence, but $\{1, 1, 2, 5, 15, 52, 203, 877\}$ us not a superincreasing sequence.

Describe an algorithm that takes as input superincreasing sequence $s_1, \ldots, s_n$ and a positive integer $k$, please find a sequnce of $s_1, \ldots, s_n$ with the sum equal to $k$. It is possible and desirable to find an algorithm that can accomplish this task in $O(n)$ time using dynamic progrmaming. If you think you've come up with an algorithm that can accomplish this task attempt to prove that it is correct.

3. For the next few problems we will explore the "placing parenthesis" problem. First I want you to compute the maximum value of the expression given that you can placeins a parenthesis between any two pair of numbers:

$$1 + 2 - 3 \times 4 - 5$$

(example of one possible ordering of parenthesis: $((((1+2)-3) \times 4)-5) = -5$)

4. How many different ordering of arithmetic operations are there in this problem? Why do we care about solving this problem with dynamic programming(find runtime of brute force)?