**CS 325 - Activity 4**

You may work in groups with up to 3 students. When submitting solutions in Gradescope select a page for each problem and the students in your group.

Written: (5 pts)

**Cookie Problem:** Suppose you are baby-sitting n children and have m ≥ n cookies to divide between them. You must give each child exactly one cookie (of course, you cannot give the same cookie to two different children). Each child has a greed factor $g_i$; 1 ≤ i ≤ n which is the minimum size of a cookie that the child will be content with; and each cookie has a size $s_j$ ; 1 ≤ j ≤ m. Your goal is to maximize the number of content children, that is the number of children i assigned a cookie j with $g_i \geq s_j$ .

a) Give a written description of a greedy algorithm to distribute cookies while maximizing the total contentment level of the children. What is your greedy criteria? Give an "informal" proof of correctness for your greedy choice.

Description:

Sort greed levels and cookie sizes, smallest to largest.

Start bottom up with the least greedy child, and that would be our greedy criteria in the algorithm, to compare children's greed to cookie size. The least greedy child will always be the first content one, if any contented mess is possible.

If greed<size, number of children satisfied + 1, else iterate cookie.

Proof:

Let i, n ∈ $\mathbb{R}$+ be arbitrary. Let $kG = \{g_1, \ldots, g_i\}$ be an non-negative integer array of kid greediness, and $cw = \{w_1, \ldots, w_i\}$ be an non-negative integer array of cookie weights. There exists an optimal combination of cookie weights $cw$, where $cw = kG_i$.

$$\Sigma_{i=0}^{n}(w_i + w_{i+n}) = kG_n$$

b) Give pseudocode for your greedy algorithm to distribute cookies.

Count = 0

Tally = 1

numchildren = length(childrenarray)

```
For (i = 1; i < numcookies + 1; i++){

    If (cookie[i] >= greed[tally]){

        Tally++

       MaxContent++}

    If (Tally == numchildren) {

         Break}

Cout >>
```

c) What is the running time of your algorithm?

2 x Merge sort O(nlogn) + algorithm O(n^2)

Code: (10 pts)

Implement your algorithms for the Cookie Problem in C++. The test cases have the following structure

```
Input:
4 // number of children
10 8 9 12 // children's greed
5 // number of cookies
11 4 6 3 9 // cookie sizes
Output:
Max contentment = 2
```

The input consists of n the number of children, followed by n greed levels, m the number of cookies and then m cookie sizes The output is the maximum number of content children.

You can use the code template provided. The name of file you submit to Gradescope must be **act4.cpp**. You may submit multiple times. Select all group member names each time you submit and also include the names of the group member in your comments.