Homework #1

(Due: 9/29/22 **)**

Group Number: 6

Group Members		
Name	SBU ID	% Contribution
Michael Lee	112424954	50
Christopher Lobato	114661869	50

CSE 373 HW #1

1. a) $\sqrt[n]{n!} = O(n)$ if positive constant exists. if $a_n = \frac{n!^{1/n}}{n}$,

$\lim_{n\to\infty} \frac{\sqrt[n]{n!}}{n} \Rightarrow \log a_n = \frac{1}{n}(\log n! - n \log n) = \frac{1}{n} \sum_{k=1}^{n} \log(k/n)$

Since $\log n! = \log 1 + \log 2 + \ldots + \log n$. Using riemann sum,

$\lim_{n\to\infty} \log a_n = \int_0^1 \log x \, dx$. So, $\lim_{n\to\infty} a_n = \frac{1}{e}$

So $\boxed{\text{true}}$ as there exists a constant c where $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) \le c$. $f(n) \le c g(n)$ for $c = 1$ and $n \ge 1$

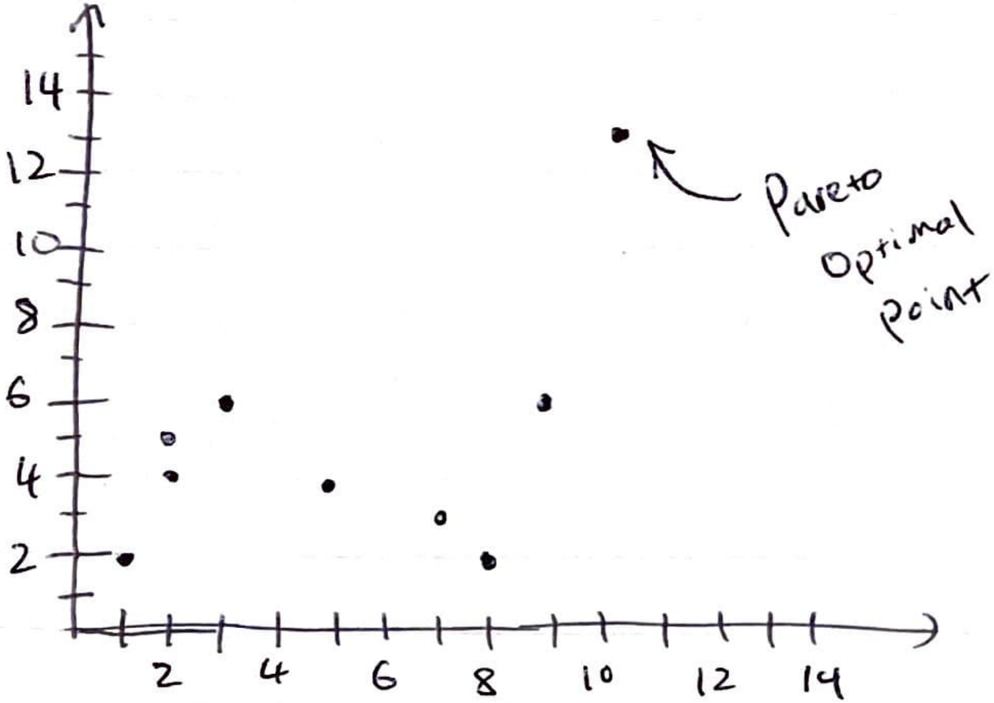
b) Using the result from above, $\lim_{n\to\infty} \frac{\sqrt[n]{n!}}{n}$ converges to $1/e$ a constant greater than 0. So, $\boxed{\text{true}}$ as there is a C where $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) \ge c$. $cg(n) \le f(n)$ for $c = 5$ and $n > 0$

c) $n|\sin(n)|$
$f(n) = n|\sin(n)|$, $g(n) = n$. Since $\sin(n)$ has a max value of 1, $f(n) \le c g(n)$ for $c \ge 1$ $\boxed{\text{true}}$

d) $f(n) = n|\sin(n)|$, $g(n) = n$. As mentioned before, $\sin(n)$ has a max value of 1. ~~$h(n) = |\sin(n)|$, $h(n) = O(1)$ as the max value is bounded by constant 1.~~
$\sin(n)$ will equal zero every pi interval. Therefore, there is no C where $cg(n) \le f(n)$ for all $n \ge n_0$ where c and n_0 must be positive. $\boxed{\text{false}}$

e) $f(n) = 1/n$, $g(n) = 1$, $\lim_{n\to\infty} \frac{1/n}{1} = 0$ Since $f(n)$ cannot be lower bounded by any $cg(n)$ where c is positive, $\boxed{\text{false}}$

2. a) Pareto optimal point for S is $(10, 13)$ as $10 > x_j$ and $13 > y_j$ for all x_j and $y_j \in S$ other than $(10, 13)$



b) The worst-case algorithm time for finding one Pareto optimal point is $O(n)$. First traverse S to find max summation value of x and y in S. The point with the max sum of x and y will be a Pareto optimal point. At worst, traverse n times through S, updating the max sum and the index of the max sum point. So, $O(n)$

$O(n) + k = O(n)$

c) The worst-case algorithm time for finding all Pareto optimal points in x-value sorted S is $O(n)$. If S is sorted in increasing order, of x, the last point will be a Pareto optimal point. The last points y value will be set as the max and all points in sorted S traversing in non-increasing x-value order will compare to current y-max. If $y_i > y_{current\ Max}$, y current max will be updated and that point will be added to Pareto optimal points set. The algorithm will traverse all of n points in S set exactly once so $O(n)$. $O(n) + k = O(n)$

d). Using merge-sort, sort the x-values of S in increasing order. Worst case of merge-sort is $O(n\log n)$. Once the S is sorted, traverse through the last element to first, adding a Pareto optimal point every time $y_i > y_{current\ max}$, traversing n times at worst. So, $O(n\log n)$

$O(n\log n) + O(n) + k = O(n\log n)$

e) Same as above, use merge-sort to first sort the array, where worst case is $O(n\log n)$. After set S is sorted traverse through S where if two consecutive x-values are met, it will take one step back, set the y current max and traverse the duplicate x-values, updating y current max and deleting all points that has been traversed with lower y values. Once a different x-value is met, the process is repeated if a new two consecutive x-values are seen. At worst, this traversing will take $O(n)$. After the set has been trimmed of duplicating x-values, repeat the steps of part (d), traversing from last to first element, adding the last element as Pareto Optimal point and every $y_i > y_{current\ max}$ being added. So, $O(n\log n)$

$O(n\log n) + O(n) + O(n) + k = O(n\log n)$

f) As done in part (e), I would sort the set S but by using quick sort as it is considered more effecient in terms of avg time complexity and space complexity. The sorted S is then traversed, deleting any duplicate x-values, trimming like part (e). Then trimmed S is then traversed from last until more than 7 Pareto optimal points are found. (This algorithm will initially check if there are more than 7 sets of points for best case).

Best: $O(1)$, Avg: $O(n\log n)$, Worst: $O(n^2)$ (as quick sort can be n^2)

3. a) $(f(n) = o(g(n))) \Rightarrow (\log(f(n)) = o(\log(g(n))))$

$\boxed{\text{False}}$ using definition of o-notation where $f(n) = n$, $g(n) = n^2$

$\lim_{n\to\infty}\left(\frac{n}{n^2}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right) = 0$. However, $\lim_{n\to\infty}\left(\frac{\log(n)}{\log(n^2)}\right)$

$= \lim_{n\to\infty}\left(\frac{\log(n)}{2\log(n)}\right) = 1/2$

b) $\boxed{\text{False}}$ if $f(n) = 2$ and $g(n) = 1$ $f(n) = O(g(n))$

but $\lim_{n \to \infty} \left(\frac{\log(f(n))}{\log(g(n))} \right) = \lim_{n \to \infty} \left(\frac{1}{0} \right)$ = undefined

c) $\boxed{\text{True}}$ by definition of $o(g(n))$, $\lim_{n \to \infty} \frac{2^{f(n)}}{2^{g(n)}} = 0$

which can be rewritten as $2^{f(n)-g(n)}$. However since we know that $\lim_{n \to \infty} \frac{2^{f(n)}}{2^{g(n)}} = 0$, $g(n)$ must be infinitely bigger than $f(n)$. Meaning $2^{f(n)-g(n)}$ will also approach zero

d) $\boxed{\text{False}}$ if $f(n)$ is $2n$ and $g(n)$ is n, the

$\lim_{n \to \infty} \frac{2^{2n}}{2^n} = \lim_{n \to \infty} 2^n = \infty$. So $2^{f(n)} \neq O(2^{g(n)})$

4. The order is: $\log(n), \sqrt{n}, n\log n, n^2, n^3, (5/4)^n, e^n, 5^n, n!,$ $(n+2022)!, 2^{2^n}$

a) checking $\log(n) = O(\sqrt{n})$ $\lim_{n \to \infty} \left(\frac{\log(n)}{\sqrt{n}} \right)$

Using L'Hopital's rule, $\lim_{n \to \infty} \left(\frac{1/x}{1/2\sqrt{x}} \right) = \lim_{n \to \infty} \left(\frac{2}{\sqrt{x}} \right) = 0$

b) $\sqrt{n} = O(n)$ $\lim_{n \to \infty} \left(\frac{\sqrt{n}}{n} \right) = \lim_{n \to \infty} \left(\frac{1}{n^{1/2}} \right) = 0$

c) $n = O(n\log n)$ $\lim_{n \to \infty} \left(\frac{n}{n\log n} \right) = \lim_{n \to \infty} \left(\frac{1}{\log n} \right) = 0$

d) $n\log n = O(n^2)$ $\lim_{n\to\infty}\left(\frac{n\log n}{n^2}\right) = \lim_{n\to\infty}\left(\frac{\log n}{n}\right)$

Using L'Hopital's rule, $\lim_{n\to\infty}\left(\frac{1/x}{1}\right) = 0$

e) $n^2 = O(n^3)$ $\lim_{n\to\infty}\left(\frac{n^2}{n^3}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right) = 0$

f) $n^3 = O((5/4)^n)$ $\lim_{n\to\infty}\left(\frac{n^3}{5/4^n}\right)$ use L'Hopitul's rule,

$\lim_{n\to\infty}\left(\frac{3n^2}{(5/4)^n\ln(5/4)}\right) \xrightarrow{\text{again}} \lim_{n\to\infty}\left(\frac{6n}{\ln^2(5/4)(5/4)^n}\right) \xrightarrow{\text{again}} \lim_{n\to\infty}\left(\frac{6}{\ln^3(5/4)(5/4)^n}\right)$

$\xRightarrow{\text{again}} \lim_{n\to\infty}\left(\frac{3\ln(2)\cdot 2^{2n+2}}{\ln^3(5/4)\cdot 5^n\ln(5)}\right) \xrightarrow{\text{again}}$ 3 more derivation

$\to \frac{3\ln^5(2)(64)}{\ln^5(5)\ln^3(5/4)} \cdot \lim_{n\to\infty}\left((4/5)^n\right) = 0$

g) $(5/4)^n = O(e^n)$ $\lim_{n\to\infty}\left(\frac{5/4^n}{e^n}\right) = \lim_{n\to\infty}\left(\frac{5/4}{e}\right)^n = 0$

h) $e^n = O(5^n)$ $\lim_{n\to\infty}\left(\frac{e^n}{5^n}\right) = \lim_{n\to\infty}\left(\frac{e}{5}\right)^n = 0$

i) $5^n = O(n!)$ $\lim_{n\to\infty}\left(\frac{5^n}{n!}\right)$ Factorial functions grow asymptotically larger than exponential functions, so limit will be 0 (can see through Stirling's approx.)

j) $n! = O(n+2022)!$ $\lim_{n\to\infty}\left(\frac{n!}{(n+2022)!}\right) = \lim_{n\to\infty}\left(\frac{1}{(n+2022)\cdots(n+1)}\right) = 0$

k) $(n+2022)! = O(2^{2^n})$ $\lim_{n\to\infty}\left(\frac{(n+2022)!}{2^{2n}}\right)$ use Stirling's approx, we can see that

2^{2n} gets marginally larger as $n\to\infty$ so 0.

Task 1. [60 Points] True or False Determine whether the following statements are true or false. Justify your answers. You can assume that n is a positive integer.

(a) $\sqrt[n]{n!} = O(n)$

We can check for the following if the above is true then $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right)\leq c$

So… $\lim_{n\to\infty}\left(\frac{\sqrt[n]{n!}}{n}\right) = \lim_{n\to\infty}\left(\frac{(n!)^{\frac{1}{n}}}{n}\right)$ using L'Hopital's rule we can see that the denominator's derivative is 1 while the numerator still approaches infinity as *n* increases. So, limit tends towards infinity Therefore $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \infty$ and the above statement is **FALSE**

(b) $\sqrt[n]{n!} = \Omega(n)$

If the above is true, then there exists a positive constant *c* such that $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right)\geq c$

$\lim_{n\to\infty}\left(\frac{\sqrt[n]{n!}}{n}\right) \approx \lim_{n\to\infty}\left(\frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^{n\frac{1}{e12n}}}{n}\right)$, similar to the previous problem we can see that after applying L'Hopital's rule that the numerator will grow much quicker than the denominator and therefore the limit approaches infinity. So above statement is **TRUE**

(c) $n|\sin sin\ (n)\ | = O(n)$

We can check for the following if the above is true then $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right)\leq c$

$\lim_{n\to\infty}\left(\frac{n|\sin sin\ (n)\ |}{n}\right) = \lim_{n\to\infty}\left(|\sin sin\ (n)\ |\right)$ In this case the statement is **FALSE** as the limit diverges and the following inequality does not hold $0\leq f(n)\leq cg(n)$ for all $n\geq n_0$ there are values of *n* in which $f(n)\geq cg(n)$

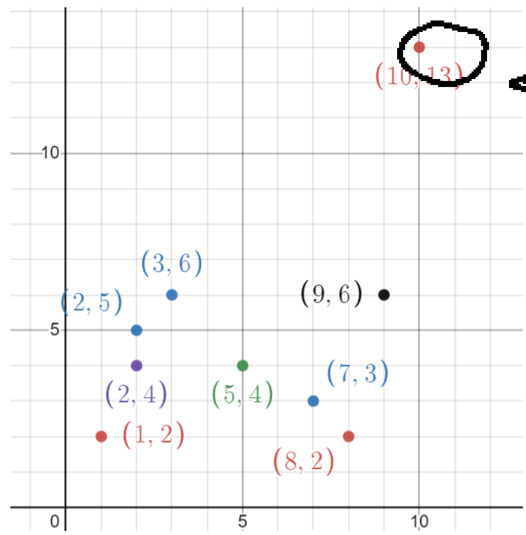
(d) $n|\sin sin\ (n)\ | = \Omega(n)$

This is **TRUE** since there exists a *c* such as $c = \frac{1}{2}$ where $cg(n)\leq f(n)$ for all $n\geq 0$

(e) $\frac{1}{n} = \Theta(1)$

$\lim_{n\to\infty}\left(\frac{\frac{1}{n}}{1}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right)$, as we can see the limit approaches 0 and this does not change no matter the *c* therefore we can conclude that the f(n) can be an upper bound but not a lower bound. Therefore the statement above is **FALSE**.

Task 2. [70 Points] Finding the Pareto optimal points Let S = {(x1, y1),(x2, y2), · · · ,(xn, yn)} be a set of n points where all coordinates are real numbers. A point (xi , yi) is called a Pareto optimal point if for every other (xj , yj) ∈ S, at least one of the following two inequalities hold:



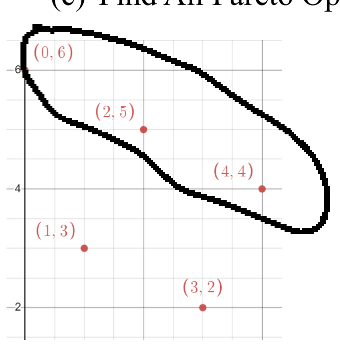
only pareto point

(a)

(b) Find One Pareto Point

```
FindAParetoPoint(S)
    rightMostPoint = S[1]
    for i = 1 to n
        current = S[i]
        if(current[0]>rightMostPoint[0])
            rightMostPoint = current
        if(current[0]==rightMostPoint[0])
            if (current[1] >rightMostPoint[1])
                rightMostPoint = current
    return rightMostPoint
```

(c) Find All Pareto Optimal Points given Sorted set S and Unique x's



Assume that the sort leaves us with the following points x's sorted in descending order

S= {(4,4), (3,2), (2,5), (1,3), (0,6)} this is preferable since we want to start checking from the right most point. Then we know that as we traverse S the following should hold $x_i > x_{i+1}$. Therefore, we just need to check the y's

```
FindAllParetos (S)
    rightMostParetoY = S[1][1]
    solution = []
    for i = 1 to n
        if (S[i][1] > rightMostParetoY)
            solution.append(S[i])
    return solution
```

(d) Find All Pareto Points given unsorted set S

```
findAllParetosUnsorted(S)
    mergeSortDescending (S)
    rightmostParetoY = S[1][1]
    solution[]
    for i = 1 to n
        if S[i][1] > rightMostParetoY
            solution.append(S[i])
    return solution
```

(e) Find all Pareto Points given no unique x's

```
findAllParetosNoUniqueX(S)
    mergeSortDescending(S)
    rightMost_x = S[1][0]
    rightMost_y = S[1][1]
    firstPareto = S[1]
    solution = []
    i = 1
    current_x = rightMost_x
```

```
        while current_x == rightMost_x
                current_x == rightMost_x
                current_x = S[i][0]
                if(points[i][1]>rightMost_y)
                        rightMost_y = S[i][1]
                        firstPareto = S[i]
                i = i+1
        paretos.append(firstPareto)
        numOfParetos = 1
        for i =1 to n
                if S[i][0] == paretos[numOfParetos][0]:
                        if point[1]> paretos[numofParetos][1]
                                paretos[numOfParetos] = S[i]
                        else if points[1] >paretos[numOfParetos][1]
                                paretos.append[S[i]]
                                numOfParetos = numOfParetos + 1
        return paretos
```

(f) Find if set has 7 Pareto Optimal Points

```
FindIf7Paretos(S)
        if S.length < 7
                return false
        solution = []
        for i = 1 to 8
                currentMax = S[1]
                found = false
                for j =1 to S.length
                        if S[j][0] > currentMax [0]
                                currentMax = S[j], found = true
```

```
            if solution is NOT Empty AND currentMax[0] == solutions[[solutions.length][0]]
And Found = True
                  if currentMax[1] > solutions[solutions.length][0]
                        solutions[solutions.length] = currentMax
            else if found== true
                  solutions.append(currentMax)
            if found == true
                  S.remove(currentMax)
      if solutions.length>=7
            return true
      else
            return false
```

Task 4. [60 Points] Sort by Growth Rate Arrange the following functions in nondecreasing order of growth rate (show necessary derivations). Assume that the 'log' function has base 2 unless explicitly mentioned otherwise

log(n),$\sqrt{n}, n, n \log log\, n$, n^2, n^3, $\left(\frac{5}{4}\right)^n$, e^n, 5^n, $n!$, $(n + 2022)!$, 2^{2^n}

(a) log(n), $\sqrt{n}$,
$\lim_{n\to\infty}\left(\frac{\log log\,(n)}{\sqrt{n}}\right)$ we can use l'hopital's and check the limit of
$\lim_{n\to\infty}\left(\frac{\frac{1}{n\ln ln\, 2}}{\frac{1}{2}n^{-1/2}}\right) = \lim_{n\to\infty}\left(\frac{\frac{1}{n\ln ln\, 2}}{2n^{1/2}}\right)$ the numerator is a constant of 1 and the denominator grows towards infinity as n approaches infinity so overall the limit approaches 0
So log $log\,(n) \; = O\left(\sqrt{n}\right)$

(b) $\sqrt{n}, n$
$\lim_{n\to\infty}\left(\frac{\sqrt{n}}{n}\right)$ we use l'hopital's rule again and check the following
$\lim_{n\to\infty}\left(\frac{\frac{1}{2}n^{-\frac{1}{2}}}{1}\right) = \lim_{n\to\infty}\left(\frac{1}{2n^{\frac{1}{2}}}\right)$ We can see that the limit approaches 0 as the denominator increases therefore $\sqrt{n} = O(n)$

(c) n, n log $log\, n$
$\lim_{n\to\infty}\left(\frac{n}{n\log log\, n}\right) = \lim_{n\to\infty}\left(\frac{1}{\log log\, n}\right)$ we know that as n increases that log(n) will approach infinity so the limit approaches 0 so we can conclude $n = O(n$ log $log\,(n)$)

(d) $n \log log\, n, n^2$

$\lim_{n\to\infty}\left(\frac{n\log log\, n}{n^2}\right) = \lim_{n\to\infty}\left(\frac{\log log\, n}{n}\right)$ we can then use l'hopital's rule and check the following $\lim_{n\to\infty}\left(\frac{1}{n\ln ln\, 2}\right)$ in this case we can see that the limit approaches 0 and can conclude that $n \log log\, n = O\left(n^2\right)$

(e) $n^2,\ n^3$

$\lim_{n\to\infty}\left(\frac{n^2}{n^3}\right) = \lim_{n\to\infty}\left(\frac{1}{n}\right)$ here we can easily see that the rate of growth of n^3 is quicker than n^2 so we can say that $n^2 = O\left(n^3\right)$

(f) $n^3, \left(\frac{5}{4}\right)^n$

$\lim_{n\to\infty}\left(\frac{n^3}{(5/4)^n}\right) \to^{LH} \lim_{n\to\infty}\left(\frac{3n^2}{(5/4)^n(5/4)}\right)$

$\to^{LH} \lim_{n\to\infty}\left(\frac{6n}{(5/4)^n \ln ln\,(5/4)\ \ln ln\,(5/4)}\right) \to^{LH} \lim_{n\to\infty}\left(\frac{6}{(5/4)^n \ln ln\,(5/4)\ \ln ln\,(5/4)\ \ln ln\,(5/4)}\right)$

After applying L'hopital's rule a few times we can then come to the conclusion that as n becomes large the limit approaches 0 and therefore $n^3 = O\left((5/4)^n\right)$

(g) $\left(\frac{5}{4}\right)^n, e^n$

$\lim_{n\to\infty}\left(\frac{(5/4)^n}{e^n}\right) = \lim_{n\to\infty}\left(\left(\frac{5}{4e}\right)^n\right)$ Here we can see that the denominator grows quicker then the numerator so the resulting limit is equal to 0 and we can conclude that $(5/4)^n = O\left(e^n\right)$

(h) $e^n, 5^n$

$\lim_{n\to\infty}\left(\frac{e^n}{5^n}\right) = \lim_{n\to\infty}\left(\left(\frac{e}{5}\right)^n\right)$ Here the inner fraction is less than 1 so the limit approaches 0 and therefore $e^n = O\left(5^n\right)$

(i) $5^n, n!$

$\lim_{n\to\infty}\left(\frac{5^n}{n!}\right) \approx \lim_{n\to\infty}\left(\frac{5^n}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12}}n}\right)$ from sterling's approximation we can see that $n!$ grows at a much quicker rate than 5^n for large inputs of n numbers so we can say $5^n = O(n!)$

(j) $n!, (n + 2022)!$

$\lim_{n\to\infty}\left(\frac{n!}{(n+2022)!}\right) = \lim_{n\to\infty}\left(\frac{n\cdot(n-1)\cdot(n\cdot 2)\ldots 1}{(n+2022)\ldots n\cdot(n-1)\ldots 1}\right) = \lim_{n\to\infty}\left(\frac{1}{(n+2022)\cdots(n+1)}\right)$ from this we can conclude that the limit approaches 0 and therefore $n! = O(n + 2022)!$

(k) $(n + 2022)!, 2^{2^n}$

$\lim_{n\to\infty}\left(\frac{(n+2022)!}{2^{2^n}}\right)$ in this case from sterling's approximation $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12}}n$ we know that this is a close approximation of n!. From this we know that 2^{2^n} grows much quicker for large n's in comparison to n!. Since (n+2022)! and n! are in a similar class of functions then 2^{2^n} must also be larger than (n+2022)! for a large n.