コンピューターのオーバーフロー回避

本の４９ページに対数を使ってオーバーフローを回避する方法が紹介されていますが、その解説です。
コンピューターの扱う数の桁数は機種により決まっているので、それより大きな数を入れるとオーバーフローというエラーを起こします。
例えば１００人の中から３人選ぶ方法は何通りあるか求める計算では

$$ {}_{100}C_3 = \frac{100!}{3! \times (100-3)!} = \frac{100 \times 99 \times 98}{3 \times 2 \times 1} = 161700 $$ （通り）……①

となりますが、この計算の途中で出てくる１００！という数は
$100! = 100 \times 99 \times 98 \times 97 \times \cdots\cdots \times 4 \times 3 \times 2 \times 1$
で、大きすぎてオーバーフローしてしまいます。
上の式の場合は分数を約分すると 100!が消えてしまうのでいいのですが、
これを計算しなければならない場合もあります。
この場合は以下のように対数をとると計算ができます。
$\log(100!) = \log 100 + \log 99 + \log 98 + \cdots + \log 3 + \log 2 + \log 1$
$= 4.605170186 + 4.59511985 + 4.584967479 + \cdots + 1.098612289 + 0.693147181 + 0$
$= 363.739375$

同様に

$\log(3!) = 1.791759$ 、 $\log(97!) = 349.9541$

$$ \log({}_{100}C_3) = \log\left(\frac{100!}{3! \times (100-3)!}\right) = \log(100!) - \log(3!) - \log(97!) $$
$= 363.739375 - 1.791759 - 349.9541$
$= 11.993498$
これより指数をとって元にもどすと

$e^{11.993498} = 161700$ (通り)となり①の計算と一致します。

ここで $\log(100!) = 363.739375$ となりオーバーフローは起きません。

そして上の式では対数をとると掛け算だったところは足し算に、割り算だったところは引き算に計算が変わっています。

ですから途中でオーバーフローするような大きな数が出てくる計算では、まず式全体を対数に変換して掛け算は足し算に、
割り算は引き算にすればオーバーフローは起きません。

しかし、式の中が掛け算と割り算だけならいいのですが、足し算や引き算があった場合には変換のしようがなくなります。
このときは以下のように計算します。

足し算の例でいうと、X＋Y を計算したいときは X も Y もそのままではオーバーフローしてしまうほど大きい数ですから
log(X+Y)としなければならず、この場合はまず logX と logY を計算してその大きさを比較します。

そしてその差が大きければ足し算をしても小さいほうの数は計算機の有効桁数からはずれてしまい、結果に影響を及ぼさなくなります。
つまり､X のほうが十分大きければ近似的に X+Y=X なのですから log(X+Y)=logX となります。

次に、X と Y の差がそれほど大きくない場合は logX と logY からそれぞれ logZ を引いた logX-logZ と logY-logZ を計算します。
ここで Z は logX-logZ と logY-logZ の指数をとって元にもどしたときにオーバーフローしないような数を選びます。

そして指数をとった $e^{(\log X-\log Z)}+e^{(\log y-\log Z)}$ を計算し、再び対数をとって logZ を加えれば求める log(X+Y)となります。