談　話　室

e の計算のこと

電子計算機ができるまえから，e だの π だののあたいは，小数点以下かなりさきのほうまでしられていた．電子計算機の時代になって，ENIAC かなんかで，機械による計算が開始されてから，計算したけたかずと，それに要した時間の記録がつぎつぎと更新されている．e や π のながいあたいも，もうありがたがられなくなってしまった．ご存知のとうりわが日本でも，4～5年まえ，電子計算機がぼつぼつとできたころ，このあまりやくにたたない計算を，あっちこっちでやっていた．東大の PC-1 での経験によると，1958年のはる，誕生してまもなくは，たいした計算がなくて，どんな program をつくっても，またたくまにおわってしまい，高橋先生をはじめ一同，なんとか計算機をながくうごかしてみたいと，いまにしておもえば，まことに珍妙な「ねがい」をもっていた．（当節は，PC-2 の先行制御による process でもいらいらする始末，うつればかわる，よのなかである．）当時，高橋研にいた program の名人，吉川君（MIT を経由して目下 Princeton）が，なにか program をかいているとおもったら，e と π を 1,000 けたもとめてご覧にいれるといいだした．吉川君がはじめるまえは，高橋研でもおおかたは e や π に関心がなかった．いまでも実はあまりない．だいたい当時，PC-1 にわり算の命令さえなかった．その PC-1 でさてやってみると，たいへんな時間ののちに無事こたえがでてきた．e は30分よりも π は1時間よりもはるかにかかっていたとおもう．こたえをみると，うへ，あってる，すげえ，すげえ．おい，もっと速度をあげるんだ，てなもんで，計算機をながくうごかしているのが「ねがい」だったのに，時間をなんとかみじかくしようと，software にくふうをこらし，わり算だの，e 計算用特別命令のおもむきをもつ quick shift なんど hardware にくみこみ，e 1,000 けたは，58 年のあき，吉川君が MIT に出発するころ，9分だったか13分だったかにまで短縮された．

おなじころ，通研の伊吹君や，電試の高橋氏のところでも e を 1,000 けたもとめ，それぞれうちではなん分でございといばっていたもので，e の 1,000 けたというのは，「標準問題」の観を呈していた．どこがなん分でふんぞりかえっていたかはすっかりわすれてしまったが，東大が一番，電試が二番，通研がわりにおそくて三番だったようだ．ある日，その通研がずい分おいついてきたとのうわさ．電試の高橋氏は，「通研は4項ずつまとめて計算する方式にしたらしい」といわれ，ぼくが，「なるほどそれは名案，東大のもそうなおそう」といったとおもう．ところがかえって吉川君のつくった program をみると，ちゃんと4項ずつまとめるようになっていたので，はじをかいた次第．

4項まとめると，1,000 けただすのに，$n=470$ くらいから $\frac{1}{n}+1)\frac{1}{n-1}\cdots\cdots$ をはじめるから，PC-1 の 36 bit には，$n\cdot n-1\cdot n-2\cdot n-3$ がおさまらない．しかし吉川君は $n\cdot n-1\cdot n-2\cdot n-3$ が4でわれることを利用して，みぎに2けた shift しておさめてしまっていた．あとで，3項ずつまとめるかわり shift はしない program をつくってみたら，shift のせいかどこのせいか，そのほうがすこしはやくて，結局4分半で計算できるようになった．こうして e の 1,000 けたは，機械の性能の test から，programming の競争になってきた．

4項まとめるとか3項まとめるとかいうのは e の計計算の途中

$$\cdots\Bigl(1+\underbrace{\frac{1}{n-3}\Bigl(1+\overbrace{\frac{1}{n-2}\Bigl(1+\frac{1}{n-1}\Bigl(1+\frac{1}{n}(1+Q)\Bigr)}^{B}}_{A}\cdots$$

の A または B を

$$\text{A}\quad (Q+((\overline{n-2}+1)\overline{n-1}+1)n+1)/\overline{n-3}\cdot\overline{n-2}\cdot\overline{n-1}\cdot n+\cdots\cdots$$

$$\text{B}\quad (Q+(\overline{n-1}+1)n+1)/\overline{n-2}\cdot\overline{n-1}\cdot n+\cdots\cdots$$

の式でやってしまうことである．

e の計算の趣味は，しいてわければ「よりはやく」種目と「よりながく」種目とになる．どちらも外国製の機械が世界記録をもっているようだが，日本記録はどうのびるかという興味はまだある．

まずはやい方．この馬鹿げた計算は，ずい分したびになったが，最近，東大勢にもういちど chance はあたえられた．PC-2 ができた去年のなつ，(つまり 1961年のなつのことである．雑誌がことし中にでますように）e の 1,000 けたが PC-2 ではどのくらいでできるかというはなしがでた．いまの program なら計算

にまず30秒だと，後藤さんと estimate した．さっそく program を PC-1 から2へほん訳してかけてみるとちょうど30秒．これに改良をくわえ，うえにのべた3項とか4項とかまとめるのを，もっと自由自在にして，2^{47} に分母がおさまるまでいれることにした．これが容易になったのは，B-register の威力である．

また高橋先生の idea で，最初から全体のながさの multiple precision をやらずに，1回目のわり算で single precision，2回目のわり算で double precision というふうに，precision をのばしていくように改善した．そのあげく，12秒で1,000けたの計算ができるようになった．いまは，わり算のかわりに逆数をかけるのはどうかとかんがえている．PC-2 では，かけ算はわり算の6倍の速度だから．

これよりはやい日本記録はまだきかない．でもあまりいいきもちではない，やがて記録はやぶられるだろうから．PC-2 よりはやい機械をつくりつつあるところもすくなくない．そこではやい機械ができれば，まず e の1,000けたはやってみるだろう．うえに公開した方法ならかなりはやくいくはず．どこがなのりでることか．

一方，ながくするこころみでは，それぞれの国産機で，どんなことをやったかきいていない．PC-1 では 2,500けたもとめた．36 bit 256 語で 2,500 けたもとめるのは特別のくふうが必要だった．まず1,000けたを計算，印刷する program ではじめの1,000けたをふつうにもとめる．つぎに 2,500けた計算する program で，2,500けた(まえの1,000けたもふくめて)もとめ，memory にいれておく．いまの program をけして，10^{1000} 倍する program をいれ，こたえを 10^{1000} 倍して，はじめの1,000けたをすて，memory に余裕をつくり，そこに1,500けた print する program をいれる．これでおわりの1,500けたを print する．計算の program より print する program の方が location をおおく必要とするので，うえのようになってしまった．PC-2 では memory がたくさんあるので，苦もなく1万けた2万けたなどもとめられる．東大(本郷)ので2万8,000けたの容量がある．1万けたの計算に，前記の program で19分かかる．ただし，print がひとしごとなのはいうまでもない．

これは外国のはなし．最近の Mathematics of Computation によると，IBM 7090 で10万けたもとめるのに，2時間だったそうで，おそれいっているところだが(はやさの記録でもあろう)，きくもなみだのものがたりは，Illiac による6万けたの計算である．これは日立（電試）の高橋氏から資料を先日みせていただいてしったので，つぎのようにして，40 bit 1,024

附 表 Wheeler による e の計算

RUN	DIVISOR D	OLD REMAINDER R(D)	QUOTIENT $Q(D):=\frac{Q(D+1)+R(D)}{D}$	NEW REMAINDER R′(D)	Q′(D):=SUM OF DIGIT IN Q(D) MOD 9	Q′(D)×D+R′(D)
1	10	1	0.1000 0000 0000	0	1	1
	9	1	0.1222 2222 2222	2	5	2
	8	1	0.1402 7777 7777	6	0	6
	7	1	0.1628 9682 5396	5	2	1
	6	1	0.1938 1613 7566	0	2	3
	5	1	0.2387 6322 7513	1	4	3
	4	1	0.3096 9080 6878	1	1	5
	3	1	0.4365 6360 2292	2	3	2
	2	1	0.7182 8180 1146	0	2	4
	1	1	1.7182 8180 1146	0	3	3
2	10	0	0.0000 0000 0000	0	0	0
	9	2	0.2222 2222 2222	2	6	2
	8	6	0.7777 7777 7777	6	3	3
	7	5	0.8253 9682 5396	5	3	8
	6	0	0.1375 6613 7566	0	2	3
	5	1	0.2275 1322 7513	1	4	3
	4	1	0.3068 7830 6878	1	1	5
	3	2	0.7689 5943 5626	0	7	3
	2	0	0.3844 7971 7813	0	8	7
	1	0	0.3844 7971 7813	0	8	8

語の計算機で6万けたをもとめたのだ．附表をご覧ねがいたい．これは program 懇談会の資料の借用だが，これでは $\frac{1}{10!}$ までの和で e がもとめてある．つまり $e=\frac{1}{10}+1)\frac{1}{9}+\cdots\cdots+1)\frac{1}{1}$ で，小数点以下24けたもとめたところだ．$\frac{1}{10\,!}$ までだから24けたまでは正確ではない．この計算機には(計算用に)memory が10進12けたまでしかないとする．この12けたに小数の商がはいっている．まず商を0にする．商に1をくわえて10でわる．できた商でまえの商をおきかえる．12けためにでた剰余（いまは0）を output tape にだす．この商に 1 をくわえる．9でわる．商をまたあたらしいのでおきかえる．剰余2を output にだす．このように，まえの商に1をくわえてはひとつずつすくないかずでそれが1になるまでわり，商をおきかえ，剰余を tape に output する．1でわるところまでいけば，第1の run はおわりで，小数点以下12けたまでもとまったことになる．これを print する．第2の run では，まず商を0にし，第1の run で1を商にくわえた手順のかわりにさっき output した tape から剰余をよんで，それをくわえてわり算をする．商をおきかえ，あたらしい output をつくるところはおなじである．Illiac ではこの方法で，run あたり5000けたずつもとめ，（1回のわり算はうえの例の10進1けたずつのかわりに10進10けたずつ，ついでに $10^{10}-1$ を法とする check もおこなった）第12の run までで6万けたをえた．所要時間は40時間．

万一これから e を計算してみようという奇特なご仁があれば，このむだばなしもおやくにたとうか．実は e の計算は，PC-1 や PC-2 では machine test program のひとつで，それにしては，すこし熱をあげすぎたときもあったとおもっている．最後に PC-2 の現在の program と，Illiac の6万けたの program の思想を Algol で（それも GO TO をつかわない流儀で）かいてみた，おひまなむきは，ご解読ください．

1. PC-2 による e 1,000けた計算の program，こたえ $e-2$ は2進小数で m[1] から m[72] にえられる．k は precision counter，j はなん項まとめて計算したかの counter である．

```
begin integer a, d, n, i, j, k, r;
integer array m[1:72];
for i:=1 step 1 until 72 do m[i]:=0;
k:=0;
for n:=470, n-j while n≥2 do
begin j:=1; d:=n; r:=1;
  for a:=n-1, a-1
    while d×a<2↑47 and a≥2 do
  begin d:=d×a; j:=j+1 end;
  for a:=n-j+1 step 1 until n-1 do
  r:=r×a+1;
  if k<72 then k:=k+1;
  for i:=1 step 1 until k do
  begin a:=r×2↑47+m[i];
    m[i]:=a÷d; r:=a-m[i]×d
  end
end end
```

2. Wheeler による e 6万けた計算の program，N は dynamic declaration ではなく $\frac{1}{N\,!}$ までとれば精度充分になるそういうNで，数字でかいてあるとおもう．また array rem は tapeのつもり．$e-2$ が print される．check の routine は省略した．

```
begin integer add, div, num, remend, run;
integer array rem[2:N], quo[1:500];
for div:=N step -1 until 2 do rem [div]:=1;
for run:=1 step 1 until 12 do
begin
  for add:=1 step 1 until 500 do quo[add]:=0;
  for div:=N step-1 until 2 do
  begin
    remend:=rem[div];
    for add:=1 step 1 until 500 do
    begin
      num:=remend×10↑10+quo[add];
      quo[add]:=num÷div;
      remend:=num-quo[add]×div
    end;
    rem[quo]:=remend
  end;
  for add:=1 step 1 until 500 do
  PRINT (quo[add])
end end
```

（小野田セメント株式会社　和田英一）

（昭和37年6月22日受付）