Taille du triangle de Pascal

J.-F. B. 2025-04-07 19:33:25

On s'intéresse à la taille minimale occupée dans la mémoire de l'ordinateur par le stockage des coefficients du binôme $\binom{n}{j}$, $0 \leqslant n \leqslant \mathrm{N}$, $0 \leqslant j \leqslant n$. Il y a une symétrie $j \leftrightarrow n-j$ qui permet de faire gagner un facteur 2, mais on l'ignore ici. On ignore aussi la taille du « word » de l'architecture et autres effets liés, par exemple en Python, au fait que chaque entier est un « objet » qui a donc plus de structure associée, et on convertit ce problème en l'estimation de :

$$\mathrm{T_N} = \sum_{0\leqslant n\leqslant \mathrm{N}} \sum_{0\leqslant j\leqslant n} \log \binom{n}{j} \tag{1}$$

Ici log est le logarithme naturel et il faudra diviser par $\log(2)$ pour le « nombre total de bits » (approximatif). On ne le fait pas encore afin de faciliter les calculs mathématiques. Avec les effets dont on a dit qu'on les ignorait, la taille « physique » réellement occupée diffère pour chaque terme au minimum par un $\mathrm{O}(1)$, donc en tout par un $\mathrm{O}(\mathrm{N}^2)$. Donc en théorie on n'a besoin de connaître ici $\mathrm{T_N}$ que à $\mathrm{O}(\mathrm{N}^2)$ près, mais comme nous sommes intéressés par les mathématiques aussi, on fera mieux évidemment...

Ce problème est très lié à celui de l'équivalent de Stirling pour $n!$ et on peut l'aborder soit en prenant comme point de départ une version précise de cet équivalent, soit en reprenant les techniques de démonstration (liées à la sommation de Euler-MacLaurin) directement. Je vais mélanger un peu des deux mais je ne veux pas faire un exposé trop long.

On veut aussi la taille occupée par seulement une ligne du triangle de Pascal, donc on va regarder pour commencer

$$\mathrm{S}_n = \sum_{0\leqslant j\leqslant n} \log \binom{n}{j} = \log \mathrm{P}_n \tag{2}$$

$$\mathrm{P}_n = \prod_{0\leqslant j\leqslant n} \binom{n}{j} = \frac{(n!)^{n+1}}{(\prod_{j=0}^{n} j!)^2} \tag{3}$$

Pour $n \geqslant 1$ on constate

$$\frac{\mathrm{P}_n}{\mathrm{P}_{n-1}} = \frac{(n!)^{n+1}}{(n-1)!^n}\frac{1}{n!^2} = \frac{n!n!^n}{(n-1)!^n}\frac{1}{n!^2} = \frac{n^n}{n!} \tag{4}$$

Ceci donne donc une jolie formule alternative pour P_n :

$$\mathrm{P}_n = \frac{1^1 2^2 \cdots n^n}{1!2!\cdots n!} \tag{5}$$

mais on va se concentrer tout d'abord sur $n^n/n!$ puisque là on a quelque chose de très connu, la formule de Stirling. Voici un énoncé qui suffira pour nous :

$$n! = \sqrt{2\pi n}\, n^n e^{-n} e^{\mu(n)} \qquad \mu(n) = \frac{1}{12n} + \mathrm{O}(n^{-3}) \tag{6}$$

Je signale au passage qu'on peut prouver que le terme $\mathrm{O}(n^{-3})$ est de la forme $\frac{-1}{360(n+\frac{1}{2}\theta_n)^3}$, $0<\theta_n<1$ et (plus facile à trouver sur internet) qu'il y a tout un développement *asymptotique* en puissances inverses impaires de n : $\sum_{k\geqslant 1}\frac{\mathrm{B}_{2k}}{(2k)(2k-1)\cdot n^{2k-1}} = 1/(12n)-1/(360n^3)+\dots$ (ce n'est pas une série convergente).

Bref on a donc, pour $n\geqslant 1$ ($\mathrm{P}_0=1$, $\mathrm{P}_1=1$) :

$$\frac{\mathrm{P}_n}{\mathrm{P}_{n-1}} = \frac{n^n}{n!} = \frac{e^n}{\sqrt{2\pi n}}e^{-\mu(n)} \tag{7}$$

$$\mathrm{P}_n = \frac{e^{n(n+1)/2}}{(\sqrt{2\pi})^n\sqrt{n!}}e^{-(\mu(1)+\dots+\mu(n))} = \frac{e^{n(n+1)/2+n/2}}{(\sqrt{2\pi})^{n+\frac{1}{2}}n^{\frac{1}{4}}n^{\frac{n}{2}}}e^{-\frac{1}{2}\mu(n)-(\mu(1)+\dots+\mu(n))} \tag{8}$$

$$\log \mathrm{P}_n = \frac{n^2}{2}-(\frac{n}{2}+\frac{1}{4})\log(n)+(1-\frac{1}{2}\log(2\pi))n-\frac{1}{4}\log(2\pi)-\frac{1}{2}\mu(n)-(\mu(1)+\dots+\mu(n)) \tag{9}$$

Comme $\mu(k)=(12k)^{-1}+\mathrm{O}(k^{-3})$, la série $\sum_{k\geqslant 1}^{\infty}(\mu(k)-(12k)^{-1})$ converge, soit $-\mathrm{C}_1$ sa somme. On a donc

$$\sum_{1\leqslant k\leqslant n}\mu(k) = \frac{1}{12}\sum_{1\leqslant k\leqslant n}\frac{1}{k}+(-\mathrm{C}_1)-\underbrace{\sum_{k>n}(\mu(k)-\frac{1}{12k})}_{=\mathrm{O}(n^{-2})} \tag{10}$$

Maintenant on rappelle que

$$\sum_{1\leqslant k\leqslant n}\frac{1}{k} = \log n+\gamma+\frac{1}{2n}+\mathrm{O}(n^{-2}) \tag{11}$$

En assemblant tous ces éléments on obtient :

$$\log \mathrm{P}_n = \frac{n^2}{2}-(\frac{n}{2}+\frac{1}{3})\log(n)+(1-\frac{1}{2}\log(2\pi))n-(\frac{1}{4}\log(2\pi)+\frac{\gamma}{12}-\mathrm{C}_1)-\frac{1}{12n}+\mathrm{O}(n^{-2}) \tag{12}$$

Il y a certainement beaucoup à dire sur la constante $\mathrm{C}=\frac{1}{4}\log(2\pi)+\frac{\gamma}{12}-\mathrm{C}_1$ qui apparaît là... Numériquement on trouve $\mathrm{C}\approx 0{,}504\,762\,9$.

On a dit dans l'introduction que la réelle taille physique (notons-là S_N^*) dans la mémoire de l'ordinateur sera $\log_2 \mathrm{P}_\mathrm{N}+\mathrm{O}(\mathrm{N})$ et que le $\mathrm{O}(\mathrm{N})$ dépend des détails de l'architecture. Tout ce qu'on peut donc dire est que le nombre de bits nécessaire est

$$\mathrm{S}_\mathrm{N}^* = \frac{\mathrm{N}^2}{2\log(2)}-\frac{\mathrm{N}}{2}\log_2(\mathrm{N})+\mathrm{O}(\mathrm{N}) \tag{13}$$

Le deuxième terme est presque à mettre dans le $\mathrm{O}(\mathrm{N})$ car la fonction $\log_2(\mathrm{N})$ croît lentement et il faut des N très grands, pour lesquels probablement il y a presque impossibilité physique d'avoir $\mathrm{N}^2/\log(4)$ bits, pour que cela ne soit plus à considérer comme un $\mathrm{O}(\mathrm{N})$.

On reprend maintenant l'étude de $\mathrm{T_N}$ par une approche directe.

$$\mathrm{T_N} = \sum_{0\leqslant n\leqslant \mathrm{N}} \mathrm{S}_n \qquad \mathrm{S}_n = \sum_{0\leqslant j\leqslant n} \log\binom{n}{j} = \log \mathrm{P}_n \tag{14}$$

On a donc, pour $n \geqslant 1$ (les sommes vides sont nulles) :

$$\begin{aligned}
\mathrm{S}_n &= \sum_{0\leqslant j\leqslant n}\Big(\sum_{1\leqslant k\leqslant n}\log k - \sum_{1\leqslant k\leqslant j}\log k - \sum_{1\leqslant k\leqslant n-j}\log k\Big) && (15)\\
&= \sum_{1\leqslant k\leqslant n}\log k(\sum_{0\leqslant j\leqslant n} 1 - \sum_{j,\, k\leqslant j\leqslant n} 1 - \sum_{j,\, 0\leqslant j\leqslant n-k} 1) && (16)\\
&= \sum_{1\leqslant k\leqslant n}\log k(n+1-(n-k+1)-(n-k+1)) && (17)\\
&= \sum_{1\leqslant k\leqslant n}(2k-n-1)\log k && (18)\\
\mathrm{T_N} &= \sum_{1\leqslant n\leqslant \mathrm{N}}\sum_{1\leqslant k\leqslant n}(2k-n-1)\log k && (19)\\
&= \sum_{1\leqslant k\leqslant \mathrm{N}}\sum_{n,\, k\leqslant n\leqslant \mathrm{N}}(2k-n-1)\log k && (20)\\
&= \sum_{1\leqslant k\leqslant \mathrm{N}}\Big((2k-1)(\mathrm{N}-k+1) - \underbrace{(\frac{\mathrm{N}(\mathrm{N}+1)}{2} - \frac{(k-1)k}{2})}_{=\frac{1}{2}(\mathrm{N}-k+1)(\mathrm{N}+k)}\Big)\log k && (21)\\
&= \sum_{1\leqslant k\leqslant \mathrm{N}}\frac{1}{2}(\mathrm{N}-k+1)(3k-2-\mathrm{N})\log k && (22)
\end{aligned}$$

On est un peu gêné par la présence explicite de N dans ce genre de sommes. De plus on anticipe qu'il y a des compensations de termes principaux. En effet la contribution multiple de N^2 est à priori négative ! Au final on sait qu'on aura du N^3 pas du $\mathrm{N}^3 \log \mathrm{N}$. Donc il nous faut des résultats assez précis sur des sommes du type $\sum_{1\leqslant k\leqslant \mathrm{N}} k^d \log k$. Le cas $d = 0$ est la formule de Stirling... Probablement les termes sous-dominants suffiront on n'a pas besoin de pousser l'analyse trop loin.

La technique de base (des méthodes plus sophistiquées utilisent la variable complexe) est la comparaison avec une intégrale. Un point de départ possible est la formule célèbre, pour f de classe C^2 :

$$\frac{1}{2}f(0) + \frac{1}{2}f(1) = \int_0^1 f(u)\mathrm{d}u + \frac{1}{2}\int_0^1 (u-u^2)f''(u)\mathrm{d}u \tag{23}$$

Pour f de la forme $f_\mathrm{N}(x) = \mathrm{P_N}(x)\log x$ et en appliquant sur les intervalles successifs

$[1,2]$, …, $[N-1,N]$, cela donne, après avoir remarqué ici $f_N(N)=(N-1)\log N$.

$$\begin{aligned}T_N=\frac{1}{2}(N-1)\log N+\int_1^N\frac{1}{2}(N-x+1)(3x-2-N)\log x\,dx\\+\frac{1}{2}\int_0^1(u-u^2)\sum_{k=1}^N f_N''(u+k)du\end{aligned}\tag{24}$$

On peut utiliser un logiciel de calcul formel pour aider aux calculs. Tout d'abord l'intégrale sur $[1,N]$ donne :

```
> interface(prettyprint=0);
1
> simplify(int((N-x+1)*(3*x-2-N)*log(x)/2,x=1..N));
-13/24+3/8*N^2+1/6*N^3-1/4*N^2*ln(N)-N*ln(N)
```

On est content car on voit que le terme principal est bien en $N^3/6$. On voit aussi que le terme sous-dominant est $-\frac{1}{4}N^2\log N$. Faisons un peu plus soigneusement avec notre Maple (si on l'a) et regardons aussi à quoi ressemble la dérivée seconde f_N'' :

```
> fN:=(N-x+1)*(3*x-2-N)*log(x)/2;
fN := 1/2*(-N^2+(4*x-3)*N-(x-1)*(3*x-2))*ln(x)
> simplify(subs(x=N,fN)/2+int(fN,x=1..N));
-1/2*N*ln(N)-1/2*ln(N)-13/24+3/8*N^2+1/6*N^3-1/4*N^2*ln(N)
> simplify(diff(fN,x$2));
1/2*(-6*ln(x)*x^2+4*N*x-9*x^2+5*x+N^2+3*N+2)/x^2
```

On trouve un terme en $\log(u+k)$ dans $f_N''(u+k)$ et clairement ça complique la gestion de l'intégrale avec $\sum_{k=1}^N f_N''(u+k)$. Mais cela contribuera seulement à un $N\log N$ au final. Aussi le terme en $N/(u+k)$ donnera du $N\log N$. Mais on a pire! On a ce terme en $N^2\sum_{k\leqslant N}(u+k)^{-2}$ et donc lui il va donner une contribution $\sim aN^2$ pour une constante a.

On pourrait s'arrêter là car on a en tout cas $T_N=\frac{1}{6}N^3-\frac{1}{4}N^2\log N+O(N^2)$ mais on voudrait tout de même être un peu plus précis (bien que ce soit ridicule car la situation dans le monde réel n'est contrôlée qu'à $O(N^2)$ près). Les contributions qui créent du $N\log N$ sont un peu pénibles car les séries $\sum_k\log(u+k)$ et $\sum_k(u+k)^{-1}$ sont divergentes. On veut les modifier en manipulant par des intégrations par parties.

On va commencer en toute généralité par $\int_0^1(u-u^2)f''(u)du$. Il est commode de modifier le point de départ pour qu'il devienne $\int_0^1(u-u^2-\frac{1}{6})f''(u)du$. En effet $\int_0^1(u-u^2-\frac{1}{6})du=0$ et l'intégration par parties est alors plus simple.

$$\frac{1}{2}\int_0^1(u-u^2)f''(u)du=\frac{f'(1)-f'(0)}{12}+\int_0^1(-\frac{1}{12}+\frac{1}{2}u-\frac{1}{2}u^2)f''(u)du\tag{25}$$

$$=\frac{f'(1)-f'(0)}{12}+\int_0^1(\frac{1}{12}u-\frac{1}{4}u^2+\frac{1}{6}u^3)f'''(u)du\tag{26}$$

On constate $\int_0^1(\frac{1}{12}u-\frac{1}{4}u^2+\frac{1}{6}u^3)\mathrm{d}u=0$ (le polynôme est une fonction impaire de $u-\frac{1}{2}$). Une nouvelle intégration par parties donne un résultat étonnamment joli :

$$\frac{1}{2}\int_0^1(u-u^2)f''(u)\mathrm{d}u=\frac{f'(1)-f'(0)}{12}-\frac{1}{24}\int_0^1 u^2(1-u)^2 f''''(u)\mathrm{d}u \tag{27}$$

La formule (24) prend donc une nouvelle forme :

$$\begin{aligned}\mathrm{T_N}=\frac{1}{2}(\mathrm{N}-1)\log\mathrm{N}+\int_1^{\mathrm{N}} f_{\mathrm{N}}(x)\mathrm{d}x+\frac{f_{\mathrm{N}}'(\mathrm{N})-f_{\mathrm{N}}'(1)}{12}\\ -\frac{1}{24}\int_0^1 u^2(1-u)^2\sum_{k=1}^{\mathrm{N}} f_{\mathrm{N}}''''(u+k)\mathrm{d}u\end{aligned} \tag{28}$$

avec

$$f_{\mathrm{N}}(x)=\frac{1}{2}(\mathrm{N}-x+1)(3x-2-\mathrm{N})\log(x). \tag{29}$$

On peut demander à Maple (ou on fait à la main) de calculer la contribution à $\mathrm{T_N}$ de tous les termes de la première ligne, et de nous dire ce que vaut f_{N}'''' :

```
> fN_1 := diff(fN,x):
> simplify(subs(x=N,fN)/2+int(fN,x=1..N)+(subs(x=N,fN_1)-subs(x=1,fN_1))/12);
1/24*(-14*N^2*ln(N)-7*N*ln(N)-11*N+10*N^3+4*N^4-6*N^3*ln(N)-2-N^2)/N
```

Ceci n'est pas très lisible (surtout avec le `/N` à la fin) voilà mieux formaté :

$$\mathrm{T_N}=\frac{\mathrm{N}^3}{6}-\frac{6\mathrm{N}^2+14\mathrm{N}+7}{24}\log(\mathrm{N})+\frac{10\mathrm{N}^2-\mathrm{N}-11}{24}-\frac{1}{12\mathrm{N}}+\textit{intégrale avec } f_{\mathrm{N}}'''' \tag{30}$$

Et pour la dérivée quatrième :

```
> simplify(diff(fN,x$4));
(3*x^2+5*x+4*N*x+3*N^2+9*N+6)/x^4
```

autrement dit :

$$f_n''''(u+k)=\frac{3}{(u+k)^2}+\frac{4\mathrm{N}+5}{(u+k)^3}+\frac{3\mathrm{N}^2+9\mathrm{N}+6}{(u+k)^4} \tag{31}$$

Il en résulte que $\sum_{k=1}^{\mathrm{N}} f_{\mathrm{N}}''''(u+k)$ est $\mathrm{O}(\mathrm{N}^2)$ uniformément en u. Ce $\mathrm{O}(\mathrm{N}^2)$ est moins bien que ce que l'on pouvait obtenir à partir de notre résultat pour $\log \mathrm{P}_n$ qui avait une erreur en $\mathrm{O}(1/n^2)$, donc en sommant de 1 à N cela donnait pour $\mathrm{T_N}$ une erreur en $\mathrm{O}(1)$! Soyons donc un peu plus précis. Pour cela l'astuce est de faire intervenir des sommes infinies (cela aurait été impossible dans l'intégrale avec f_{N}'') :

$$\sum_{k=1}^{\mathrm{N}} f_{\mathrm{N}}''''(u+k)=\sum_{k=1}^{\infty} f_{\mathrm{N}}''''(u+k)-\sum_{k>\mathrm{N}} f_{\mathrm{N}}''''(u+k) \tag{32}$$

Le premier terme (la série infinie) est de la forme $N^2\alpha(u) + N\beta(u) + \gamma(u)$ avec des fonctions continues de u et toutes les trois indépendantes de N. Le reste $\sum_{k>N}$ est uniformément en u un $O(N^{-1})$ comme on voit en utilisant $\sum_{k>N} k^{-d} = O(N^{1-d})$ pour $d > 1$. Donc après intégration sur $[0,1]$, on obtient en tout $N^2\lambda + N\mu + \nu + O(N^{-1})$ pour certains constantes λ, μ, ν. C'est mieux qu'un grossier $O(N^2)$ mais les constantes restent assez inconnues à ce stade ! (on peut les écrire sous la forme de certaines intégrales).

La conclusion est donc :

$$T_N = \frac{N^3}{6} - \frac{6N^2 + 14N + 7}{24}\log(N) + XN^2 + YN + Z + O(N^{-1}) \tag{33}$$

avec des constantes X, Y, Z inconnues. On va pouvoir récupérer X et Y en calculant $T_N - T_{N-1}$ et en comparant à $\log P_N$ (we expect the $O(N^{-1})$ to be a $a/N + O(N^{-2})$ and contribute only to $O(N^{-2})$ in the difference).

```
> T:=N->N^3/6 - (6*N^2+14*N+7)/24*log(N) + X*N^2 + Y*N + Z:
> asympt(T(N)-T(N-1),N,2);
1/2*N^2+(-3/4-1/2*ln(N)+2*X)*N-1/3*ln(N)-1/24-X+Y-1/12/N+O(1/N^2)
```

En comparant avec (12) on trouve que c'est compatible et que $-\frac{3}{4} + 2X = 1 - \frac{1}{2}\log(2\pi)$, $-\frac{1}{24} - X + Y = -C$. Ce qui permet d'écrire X et Y si l'on veut...

$$X = \frac{7}{8} - \frac{1}{4}\log(2\pi) \approx 0{,}415\ 531 \tag{34}$$

$$Y = \frac{11}{12} - \frac{1}{4}\log(2\pi) - C \approx -0{,}047\ 565\ 5 \tag{35}$$

On se dit au final que comme on avait (12) avec une erreur en $O(n^{-2})$ on aurait sans doute pu obtenir presque aussi bien que (33) plus facilement en appliquant la même méthode de comparaison d'une somme avec une intégrale à partir de (12) et on aurait évité les complications avec un N ou N^2 venant multiplier la fonction. Le seul problème c'est qu'on aurait eu un $O(1)$ au lieu de $Z + O(N^{-1})$.

Testing numerically with $N \in \{125, 250, 500, 1000\}$ things seem to be OK and we find Z to be $\approx -0.467{,}7$. We only $C \approx 0.504{,}762{,}9$ hence for Y too we only know approximately. It gets multiplied by 1000 for $N = 1000$, so we are lucky already if we got the four displayed decimals of Z correctly. The error in $O(N^{-1})$ which we neglected is perhaps not that small for $N = 1000$. We postulated that the $Z + O(N^{-1})$ is $Z + a/N + b/N^2 + c/N^3 + \ldots$ and did some Richardson extrapolation on the differences between exact T_N and approximated one to get a refined estimate of Z.

We do feel some confidence that we got X right as the approximate formula for $N = 1000$ works well despite X being multiplied by 1000^2 which is much larger than the measured differences. And T_{1000} is about $1.653{,}512 \times 10^8$ and it is rather convincing we get an error only about equal to -0.468. We thus feel we probably got the $P(N)\log N$ terms correctly. Probably the Z rounds to -0.468 but saying it is $-0.467{,}7\ldots$ is a bit daring with the very limited numerical exploration we did.