Cœfficients binomiaux

On peut vouloir calculer $\binom{n}{k}$ en utilisant la formule du triangle de Pascal :

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Cela donne la fonction naïve :

OCaml

```
let rec binom k n =
    if k < 0 || k > n then (* éventuellement *)
        0
    else if k == 0 || k == n then
        1
    else
        binom (k - 1) (n - 1) + binom k (n - 1)
;;
```

On a alors, comme nombre d'appels à la fonction, $N_k^n = 1$ si $k = 0$ ou $k = n$ et

$$N_k^n = N_{k-1}^{n-1} + N_k^{n-1} + 1$$

sinon.

C'est une récurrence de la même forme que celle de $\binom{n}{k}$, on cherche a et b tels que $N_k^n = a\binom{n}{k} + b$ et on trouve que $N_k^n = 2\binom{n}{k} - 1$ pour $0 \leqslant k \leqslant n$.

Pour un n donné, la valeur maximale est $N_{\lceil \frac{n}{2} \rceil}^n$ et la formule de Stirling nous dit qu'elle est de l'ordre de $\frac{2^n}{\sqrt{n}}$. Pas très réjouissant !

Une meilleure idée serait d'utiliser les formules $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$ et $\binom{n}{k} = \binom{n}{n-k}$:

OCaml

```
let rec binom k n =
    if k < 0 || k > n then
        0
    else if k == 0 or k == n then
        1
    else if n - k < k then (* petite optimisation supplémentaire. *)
        binom (n - k) n
    else
         (n * binom (k - 1) (n - 1)) / k
;;
```

avec une complexité linéaire cette fois. (Attention au piège sur le `/ k`...)

☞ Remarque 1

On verra aussi une autre méthode de programmation dynamique consistant à calculer les lignes du triangle de Pascal les unes après les autres.