Discrete Mathematics and Theoretical Computer Science DMTCS vol. **13:3**, 2011, 1–8

Digital search trees with m trees: Level polynomials and insertion costs

Helmut Prodinger[1†]

[1] *Department of Mathematics, University of Stellenbosch 7602, Stellenbosch, South Africa*

received 23rd June 2011, accepted 23rd June 2011.

To Philippe Flajolet, who revisited Digital Search Trees and understood the power of Rice's integrals.

We adapt a novel idea of Cichon's related to Approximate Counting to the present instance of Digital Search Trees, by using m (instead of one) such trees. We investigate the level polynomials, which have as coefficients the expected numbers of data on a given level, and the insertion costs. The level polynomials can be precisely described, thanks to formulæ from q-analysis. The asymptotics of expectation and variance of the insertion cost are fairly standard these days and done with Rice's method.

Keywords: Digital search tree, level polynomial, Rice's method, cancellations, q-analysis.

1 Introduction

Digital search trees (=DSTs), level polynomials and insertion costs are certainly classical subjects: Knuth (1973); Flajolet and Sedgewick (2009); Mahmoud (1992); Flajolet and Sedgewick (1986); Louchard (1987); Kirschenhofer and Prodinger (1988); Szpankowski (1991); Prodinger (1995).

The following sentences from our own Prodinger (1995), which never appeared in a proper journal, can be reproduced here almost verbatim:

A DST is constructed like a binary search tree, but the decision to go down to the left or right is done according to the representation of the key as a binary string of bits. If the first bit is 0, the item goes to the left, otherwise to the right. Then the second bit is responsible for left or right, etc., until there is an empty node where the item can be stored. Decisions are made independently.

The classic book Knuth (1973) contains a more elaborate description.

Now, in order to study the average search costs in a DST built from n random data (i.e., in every decision, 0 and 1 is equally likely), the polynomial $H_n(u)$, which has as the coefficient of u^k the expected number of nodes on this level, is studied. By convention, the root is at level 0.

The formula for the *level polynomials*

$$H_n(u) = \sum_{k=1}^{n} \binom{n}{k} (-1)^{k-1} (u)_{k-1}, \qquad n \geq 1, \quad H_0(u) = 0,$$

†Email: `hproding@sun.ac.za`.

appears already in Knuth's book (Knuth, 1973, p. 504), compare also (Mahmoud, 1992, Section 6).

Here, we use the (classic) notation $(u)_k := (1-u)(1-uq)\dots(1-uq^{k-1})$; for applications to DSTs, we take $q=\frac{1}{2}$. We also need the infinite product $(x)_\infty := (1-x)(1-xq)\dots$.

Guy Louchard (1987) gave explicit expressions for the coefficients of $H_n(u)$; see also (Mahmoud, 1992, Section 6). We write $[u^k]H_n(u)$ for the coefficient of u^k in $H_n(u)$, and similarly for generating functions in 2 variables.

DSTs and Approximate Counting are very similar when it comes to analysis, see for instance Prodinger (1992). In 2011, Cichon (Cichoń and Macyna (2011)) had a novel idea about Approximate Counting, by introducing an additional parameter m, see also Prodinger (2011).

Translating this idea to the world of DSTs goes as follows: Instead of keeping one DST, we keep m DSTs, and an incoming data is attached to one of them. For algorithmic purposes (insertion and searching), this must be deterministic, but for the analysis we assume that a random DST is chosen with probability $\frac{1}{m}$. The meaning of the level polynomial is then obvious: the coefficient of u^k is the expected number of data on level k in all m DSTs combined.

Considering m-DSTs is equivalent (by adding an extra root) to the investigation of a tree structure with a prescribed root degree. This is not uncommon in Combinatorics and Computer Science; we just give one citation as an illustration: Kemp (1980).

2 An explicit formula for the level polynomials

We start from the classic formula (derived e. g., in Prodinger (1995)) for the level polynomials for classical DSTs $h_n(u)$

$$h_n(u) = \sum_{k=1}^{n} \binom{n}{k} (-1)^{k-1} (u)_{k-1}$$

and note that $h_n(1) = n$. Then the level polynomials $H_N(u)$ for the m-version satisfy

$$H_N(u) = m^{-N} \sum_{n_1+\dots+n_m=N} \binom{N}{n_1,\dots,n_m} \Big(h_{n_1}(u) + \dots + h_{n_m}(u) \Big),$$

which is easy to see since a total of N data splits into $n_1,\dots,n_m$ data each, building the m DSTs, and the individual level polynomials have to be added. The final explicit formula for the coefficients of $H_N(u)$ appears at the end of the following computations in Theorem 2.1. We can simplify:

$$\begin{aligned}
H_N(u) &= m^{1-N} \sum_{n_1+\dots+n_m=N} \binom{N}{n_1,\dots,n_m} h_{n_1}(u) \\
&= m^{1-N} \sum_{n=0}^{N} \binom{N}{n} h_n(u)(m-1)^{N-n} \\
&= m^{1-N} \sum_{n=1}^{N} \binom{N}{n} \sum_{k=1}^{n} \binom{n}{k} (-1)^{k-1} (u)_{k-1} (m-1)^{N-n} \\
&= m^{1-N} \sum_{k=1}^{N} \binom{N}{k} (-1)^{k-1} (u)_{k-1} \sum_{n=k}^{N} \binom{N-k}{n-k} (m-1)^{N-n}
\end{aligned}$$

$$= \sum_{k=1}^{N} \binom{N}{k} (-1)^{k-1} (u)_{k-1} m^{1-k}.$$

So the difference is just this extra factor m^{1-k}. Note again that for our DST application we have $q = \frac{1}{2}$, although the computations hold for general q.

Now we form a bivariate generating function:

$$\begin{aligned}
H(u,x) &= \sum_{N\ge 1} H_N(u)x^N = \sum_{N\ge 1}\sum_{k=1}^{N} \binom{N}{k} (-1)^{k-1}(u)_{k-1} m^{1-k} x^N \\
&= \sum_{k\ge 1}\sum_{N\ge k} \binom{N}{k} (-1)^{k-1}(u)_{k-1} m^{1-k} x^N \\
&= \sum_{k\ge 1} (-1)^{k-1}(u)_{k-1} m^{1-k} \frac{x^k}{(1-x)^{k+1}} \\
&= \frac{x}{(1-x)^2} \sum_{k\ge 0} (-1)^{k}(u)_{k} m^{-k} \frac{x^k}{(1-x)^{k}} \\
&= \frac{x}{(1-x)^2} \sum_{k\ge 0} (u)_k z^k \qquad \text{with} \qquad z = \frac{x}{m(x-1)}.
\end{aligned}$$

We need the classic transformation of Heine Andrews (1976):

$$\sum_{n\ge 0} \frac{(a)_n (b)_n t^n}{(q)_n (c)_n} = \frac{(b)_\infty (at)_\infty}{(c)_\infty (t)_\infty} \sum_{n\ge 0} \frac{(c/b)_n (t)_n b^n}{(q)_n (at)_n}$$

and the corollary

$$\sum_{n\ge 0} (y)_n z^n = (y)_\infty \sum_{n\ge 0} \frac{y^n}{(q)_n (1-zq^n)},$$

which is obtained by setting $a = q$, $b = y$, $c = 0$, and $t = z$ and noticing that $\dfrac{(qz)_\infty}{(z)_\infty} = \dfrac{1}{1-z}$ and $\dfrac{(z)_n}{(qz)_n} = \dfrac{1-z}{1-zq^n}$. Now we can continue:

$$\begin{aligned}
H(u,x) &= \frac{x}{(1-x)^2} (u)_\infty \sum_{n\ge 0} \frac{u^n}{(q)_n (1-zq^n)} \\
&= \frac{x}{(1-x)^2} (u)_\infty \sum_{n\ge 0} \frac{u^n}{(q)_n (1-\frac{x}{m(x-1)}q^n)} \\
&= \frac{x}{(1-x)} (u)_\infty \sum_{n\ge 0} \frac{u^n}{(q)_n (1-x+\frac{x}{m}q^n)} \\
&= m(u)_\infty \sum_{n\ge 0} \frac{(u/q)^n}{(q)_n} \left[\frac{1}{1-x} - \frac{1}{1-x(1-\frac{q^n}{m})} \right]
\end{aligned}$$

$$= m\frac{1}{(1-x)(1-u/q)} - m(u)_\infty \sum_{n\ge 0} \frac{(u/q)^n}{(q)_n} \frac{1}{1-x(1-\frac{q^n}{m})}.$$

We can read off coefficients:

$$[x^N]H(u,x) = m[x^N]\frac{1}{(1-x)(1-u/q)} - m(u)_\infty [x^N] \sum_{n\ge 0} \frac{(u/q)^n}{(q)_n} \frac{1}{1-x(1-\frac{q^n}{m})}$$
$$= m\frac{1}{1-u/q} - m(u)_\infty \sum_{n\ge 0} \frac{(u/q)^n}{(q)_n}\Big(1-\frac{q^n}{m}\Big)^N$$

and further

$$[x^N u^l]H(u,x) = mq^{-l} - m[u^l](u)_\infty \sum_{n\ge 0} \frac{(u/q)^n}{(q)_n}\Big(1-\frac{q^n}{m}\Big)^N$$
$$= mq^{-l} - m\sum_{k=0}^{l} \frac{(1/q)^k}{(q)_k}\Big(1-\frac{q^k}{m}\Big)^N [u^{l-k}](u)_\infty$$
$$= mq^{-l} - m\sum_{k=0}^{l} \frac{q^{-k}}{(q)_k}\Big(1-\frac{q^k}{m}\Big)^N \frac{(-1)^{l-k}q^{\binom{l-k}{2}}}{(q)_{l-k}}.$$

This is the explicit formula that we wanted to derive. In it, we used an expansion due to Euler:

$$(u)_\infty = \sum_{n\ge 0} \frac{(-1)^n u^n q^{\binom{n}{2}}}{(q)_n}.$$

Theorem 2.1 *The expected number of data on level l in an m-DST built from N random data is explicitly given by*

$$mq^{-l} - m\sum_{k=0}^{l} \frac{q^{-k}}{(q)_k}\Big(1-\frac{q^k}{m}\Big)^N \frac{(-1)^{l-k}q^{\binom{l-k}{2}}}{(q)_{l-k}}.$$

The instance $m = 1$ has been known before, see Louchard (1987); Prodinger (1995); Mahmoud (1992).

3 Insertion cost

The quantity $\dfrac{H_N(u)}{N}$ is the probability generating function of a random variable called *insertion cost*. While it is well studied in the classical instance, we will provide here average and variance for general (fixed) m, both, explicitly and asymptotically. (The impatient reader can already jump forward to Theorem 3.1 for the results.) We need

$$H_N'(1) = \sum_{k=2}^{N} \binom{N}{k}(-1)^k Q_{k-2} m^{1-k},$$

$$H''_N(1) = 2\sum_{k=2}^{N}\binom{n}{k}(-1)^{k-1}Q_{k-2}T_{k-2}m^{1-k}$$

with $T(k) := \sum_{j=1}^{k}\frac{1}{2^j-1}$. We use the notation $Q_n = \prod_{k=1}^{n}(1-2^{-k})$, common in Computer Science, as well as $(q)_n$.

Now we can engage into the asymptotics, using Rice's method. This method has been described in Flajolet and Sedgewick (1995):

An alternating sum can be written as a contour integral:

$$\sum_{k=2}^{N}\binom{N}{k}(-1)^k f(k) = \frac{1}{2\pi\mathbf{i}}\int_{\mathcal{C}}\frac{(-1)^N N!}{z(z-1)\dots(z-N)}f(z)dz.$$

Here, the positively oriented curve $\mathcal{C}$ enclosed the poles $2, 3, \dots, N$, and no others. This formula follows from simple residue calculations. Note also that

$$\frac{(-1)^N N!}{z(z-1)\dots(z-N)} = -\frac{\Gamma(N+1)\Gamma(-z)}{\Gamma(N+1-z)}.$$

Extending the curve of integration, we encounter extra residues; in order to keep the formula correct, these residues must be subtracted. They give us the terms of the asymptotic expansion of interest. There is in all our examples a pole at $z = 1$, and it will give us the dominant contribution.

Neglecting tiny oscillations, we can write in a suggestive way:

$$\sum_{k=2}^{N}\binom{N}{k}(-1)^k f(k) \sim \operatorname{Res}_{z=1}\frac{\Gamma(N+1)\Gamma(-z)}{\Gamma(N+1-z)}f(z).$$

For the expected value, we use

$$f(z) = (q)_{z-2}m^{1-z},$$

with

$$(q)_z := \frac{(q)_\infty}{(q^{z+1})_\infty}.$$

For convenience, we use $w = z - 1$, so that the expansions are around $w = 0$.

Just recently in our paper Prodinger (2011) we used (with $Q = \frac{1}{q} = 2$ and $L = \log Q = \log 2$)

$$Q_{w-1} = \frac{Q_w}{1-2^{-w}} \sim \Big[1 - L\alpha w + \frac{L^2}{2}(\alpha^2+\alpha+\beta)w^2\Big]\frac{1}{1-2^{-w}}$$
$$\sim \frac{1}{Lw} + \frac{1}{2} - \alpha + \frac{L}{2}\Big(\alpha^2+\beta+\frac{1}{6}\Big)w.$$

Here,

$$\alpha = \sum_{j\ge1}\frac{1}{2^j-1} \quad \text{and} \quad \beta = \sum_{j\ge1}\frac{1}{(2^j-1)^2}.$$

So we must consider

$$(q)_{w-1}m^{-w}\frac{\Gamma(N+1)\Gamma(-w-1)}{\Gamma(N-w)};$$

the residue can be computed by a computer:

$$N\Big(\log_2 N-\log_2 m+\frac{1}{2}-\alpha+\frac{\gamma-1}{L}\Big)+O(1).$$

Now, we have traditionally

$$T_{z-2}=T_{w-1}=\alpha-\frac{1}{2^w-1}-\sum_{j\ge1}\frac{1}{2^{j+w}-1}\sim-\frac{1}{Lw}+\frac{1}{2}-\frac{Lw}{12}+L(\alpha+\beta)w.$$

For the second factorial moment we use

$$f(z)=-2Q_{z-2}T_{z-2}m^{1-z},$$

so that we have to expand

$$-2(q)_{w-1}T_{w-1}m^{-w}\frac{\Gamma(N+1)\Gamma(-w-1)}{\Gamma(N-w)}$$

around $w=0$ and compute the residue, which we don't display in full:

$$N\Big[\log_2^2 N+2\log_2 N\cdot\Big(\frac{\gamma-1}{L}-\alpha-\log_2 m\Big)+\textit{further terms}\Big].$$

When we compute the variance via

$$\frac{H_N''(1)}{N}+\frac{H_N'(1)}{N}-\Big(\frac{H_N'(1)}{N}\Big)^2,$$

there are many cancellations. Altogether we summarize our results.

Theorem 3.1 *Expectation and variance of the parameter* insertion cost in m-DSTs *admit the asymptotic expansions*

$$\textit{expectation}\sim\log_2 N-\log_2 m+\frac{1}{2}-\alpha+\frac{\gamma-1}{L}+\delta_1(\log_2 N),$$
$$\textit{variance}\sim\frac{\pi^2}{6L^2}-\alpha-\beta+\frac{1}{12}+\frac{1}{L^2}+\delta_2(\log_2 N),$$

where $\delta_1(x)$ and $\delta_2(x)$ are tiny fluctuating functions that we did not compute explicitly here.

The computation of the fluctuating functions is not difficult, and very similar computations have appeared in the relevant literature many times; here is an incomplete list of references: Kirschenhofer and Prodinger (1988); Kirschenhofer and Prodinger (1991); Flajolet and Sedgewick (1995). Note that the variance is (asymptotically) a constant (plus a tiny fluctuation) which does *not* depend on m; $\delta_1(x)$ also does not depend on m. For $m=1$, this result appeared already in Kirschenhofer and Prodinger (1988); Szpankowski (1991).

So, as far as the main terms in the asymptotics are concerned, the dependency on m of average and variance is very minor, and m-DSTs don't show any improved behaviour. But such a statement can only be made after some thorough analysis, which is, why we provided it here.

References

G. Andrews. *The Theory of Partitions*, volume 2 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1976.

J. Cichoń and W. Macyna. Approximate counters for flash memory. *17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Toyama, Japan*, 2011.

P. Flajolet and R. Sedgewick. Digital search trees revisited. *SIAM Journal on Computing*, 15:748–767, 1986.

P. Flajolet and R. Sedgewick. Mellin transforms and asymptotics: Finite differences and Rice's integrals. *tcs*, 144:101–124, 1995.

P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, Cambridge, 2009.

R. Kemp. The average height of r-tuply rooted planted plane trees. *Computing*, 25(3):209–232, 1980.

P. Kirschenhofer and H. Prodinger. Further results on digital search trees. *Theoretical Computer Science*, 58:143–154, 1988.

P. Kirschenhofer and H. Prodinger. Approximate counting: An alternative approach. *RAIRO Theoretical Informatics and Applications*, 25:43–48, 1991.

D. E. Knuth. *The Art of Computer Programming*, volume 3: Sorting and Searching. Addison-Wesley, 1973. Second edition, 1998.

G. Louchard. Exact and asymptotic distributions in digital and binarysearch trees. *RAIRO Theoretical Informatics and Applications*, 21:479–495, 1987.

H. Mahmoud. *Evolution of Random Search Trees*. John Wiley, 1992.

H. Prodinger. Hypothetic analyses: Approximate counting in the style of Knuth, path length in the style of Flajolet. *Theoretical Computer Science*, 100:243–251, 1992.

H. Prodinger. Digital search trees and basic hypergeometric functions. *EATCS Bulletin*, 56:112–115, 1995.

H. Prodinger. Approximate counting with m counters: a detailed analysis. *submitted*, 2011.

W. Szpankowski. A characterization of digital search trees from the successful search viewpoint. *Theoretical Computer Science*, 85:117–134, 1991.