On the average cost of insertions on random relaxed K-d trees*

Amalia Duch† Conrado Martínez†

Abstract

In this work we refine the average case analysis of randomized insertions and deletions in random relaxed K-d trees, first given by Broutin *et al.* in [3]. The analysis is based in the analysis of the `split` and `join` algorithms, which recursively call one another and are the basis of the randomized update operations under consideration.

For $K = 2$ the average cost of insertions and deletions is $\Theta(\log n)$. For $K > 2$, this average cost is $\Theta(n^{\phi(K)-1})$, for some $\phi(K) > 1$. This immediately follows from the analysis of the expected cost s_n of splitting a tree of size n, which is the same as the expected cost m_n of joining a pair of trees with total size n. These costs are, for $K = 2$, $s_n = m_n = \Theta(n)$ and, for $K > 2$, $s_n = m_n = \Omega(n^{\phi(K)})$. In this abstract we find a closed form for the value of the exponent $\phi(K)$, as well as the constant factor multiplying the main order term in s_n.

1 Introduction

Relaxed K-d trees were introduced by the authors and Estivill-Castro in [6], as a simple alternative to standard K-d trees [2]. Both standard and relaxed K-d trees are binary trees where each of their nodes store a multidimensional key $x = (x_0, \ldots, x_{K-1})$ and a *discriminant* i, $0 \leq i < K$, so that all keys $u = (u_0, \ldots, u_{K-1})$ in the left subtree have its ith coordinate $u_i \leq x_i$, and all keys $v = (v_0, \ldots, v_{K-1})$ in the right subtree have $x_i < v_i$.

The difference between standard and relaxed K-d trees (and other variants of K-d trees such as *squarish* K-d trees [5]) lies in the way in which the discriminants are assigned to nodes. In standard K-d trees, discriminants are cyclically assigned according to level, with the root at level 0 discriminating w.r.t. the 0th coordinate. Thus a node at level j discriminates w.r.t. the ($j \mod K$)th coordinate, and there is actually no need to explicitly store the discriminants at the nodes.

In relaxed K-d trees, discriminants are assigned independently at random, with identical probability. This flexibility allows for relatively simple algorithms to reorganize and rebalance these trees. Duch *et al.* [6] designed randomized insertion and deletion algorithms which guarantee that the resulting tree is random, irrespective of the order of insertions or deletions, and any correlation present in the data. The price to pay for the flexibility of relaxation is a slightly increased expected cost of several operations, such as partial match searches [9, 6, 11] and orthogonal range searches [4, 7]. As shown in [3] and in this extended abstract, the cost of insertions and deletions is not logarithmic for $K \geq 3$, which is a more serious drawback in highly dynamic applications.

The insertion and deletion algorithms of Duch *et al.* [6] are based upon two other algorithms, the `split` and the `join` algorithms. The `split` algorithm "cuts" a given relaxed K-d tree T along a given coordinate i and a value x, returning a pair of relaxed K-d trees $\langle T^-, T^+\rangle$, with T^- containing all keys u of T such that $u_i \leq x$, and T^+ containing all the keys v such that $x < v_i$. On the other hand, the `join` algorithm blends into a single relaxed K-d tree T two relaxed K-d trees L and R such that for some coordinate i, all keys u in L satisfy $u_i \leq v_i$ for any key v in R.

The insertion algorithm works as follows: to insert a new item we follow a path down the tree with the standard algorithm, but at each step we make a random choice: whether to continue down the tree or to perform an insertion-at-root. The insertion at root is then trivially accomplished using `split`. In a deletion, we search for the item to be deleted following the corresponding path from the root, and when the sought item is found its subtrees are `join`ed.

The first analysis of the `split` and `join` algorithms appears in a recent paper by Broutin *et al.* [3]. They have shown that the expected cost s_n of splitting a tree of size n is the same as the expected cost m_n of joining a pair of trees with total size n. Also they have shown that for $K = 2$, $s_n = m_n = \Theta(n)$ and that for $K > 2$, $s_n = m_n = \Omega(n^{\phi(K)})$ for some $\phi(K) > 1$. As an immediate consequence, the average cost of insertions

*This research was supported by the Spanish Min. of Science and Technology project TIC2002-00190 (AEDRI II) and TIN2005-05446 (ALINEX). Part of this research was done while the first author was on sabbatical leave in Carleton University (Canada).

†Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. E-08034 Barcelona, Spain. {duch,conrado}@lsi.upc.es.

and deletions is $\Theta(\log n)$ for $K = 2$ and for $K > 2$, the average cost is $\Omega(n^{\phi(K)-1})$.

In this abstract we present a refined analysis of `split` and `join`, giving precise asymptotic estimates of their average cost. In particular, we find a closed form for the value of the exponent $\phi(K)$, as well as the constant factor multiplying the main order term in s_n. From there, we also obtain precise asymptotic estimates of the average cost of insertions and deletions.

2 Setting up the recurrences

Let S_n be the cost of splitting a random relaxed K-d tree of size n, and let s_n denote its expected value. Similarly, let M_n and m_n be the cost and the expected cost of joining a pair of trees of total size n. We will measure these costs as the number of visited nodes.

Consider now s_n. To start with, if $n > 0$ we must pay 1 for the visit to the root. Then, with probability $1/K$ the root of the tree discriminates by the same coordinate as the one given, so the algorithm needs to proceed recursively into the appropriate subtree. If the left subtree has size j (this occurs with probability $1/n$ for all j, $0 \le j < n$), then the split will continue in the left subtree with probability $(j+1)/(n+1)$, and it will go to the right subtree with probability $(n-j)/(n+1)$ (see Figure 1).

On the other hand, with probability $(K-1)/K$ the tree will be cut along a different coordinate to that used to discriminate at the root, say i'; hence, we will incur here the cost of splitting both subtrees L and R of T, and then joining the appropriate "pieces": either L^- and R^- or L^+ and R^+, depending on whether the root has a value smaller or greater than x for the ith coordinate. We don't know the size of the trees involved in the `join`, but this is independent of the size of L and R. And the root of T will not be involved in the `join`, that's for sure (see Figure 2).

Putting everything together, we have $s_0 = 0$ and for $n > 0$,

$$s_n = 1 + \frac{2}{nK} \sum_{0 \le j < n} \frac{j+1}{n+1} s_j + \frac{2(K-1)}{nK} \sum_{0 \le j < n} s_j + \frac{K-1}{K} \sum_{0 \le j < n} \pi_{n,j} m_j,$$

where $\pi_{n,j}$ is the probability that the total size of the trees involved in the `join` is j. Notice that we have used symmetry to simplify the first two sums in the right-hand side of this recurrence. To complete the recurrence, notice that if, for instance, we have to join two subtrees with total size j that will constitute a right subtree, and the left subtree had then size $n-j$, then there is one node to be inserted with probability

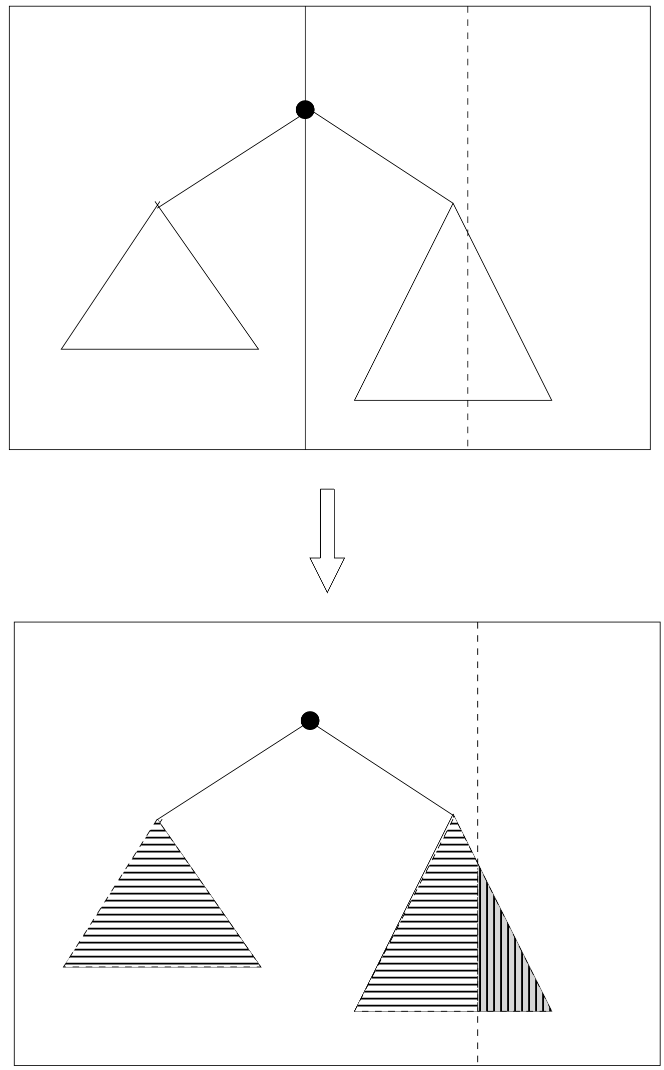

Figure 1: Split algorithm (case 1).

$(n-j)/(n+1)$. Hence, the last term in the recurrence is actually

$$\frac{K-1}{nK}\sum_{0\le j<n}\frac{n-j}{n+1}m_j+\frac{j+1}{n+1}m_{n-1-j}.$$

Therefore we have, for $n>0$,

$$s_n = 1+\frac{2}{nK}\sum_{0\le j<n}\frac{j+1}{n+1}s_j+\frac{2(K-1)}{nK}\sum_{0\le j<n}s_j + \frac{2(K-1)}{nK}\sum_{0\le j<n}\frac{n-j}{n+1}m_j, \tag{2.1}$$

Let us consider now the recurrence describing the average cost of `join`. Again, whenever $n>0$ we will visit the root of one of the two trees at least. So we pay 1. Then, with probability $1/K$ the splitting hyperplane is parallel to the discriminating coordinate of the root selected at random, say that of L, and thus we only need to join the right subtree of L with R (analogously, if the selected root were that of R, we would join the left subtree of R with L). Thus this is as if we were inserting some node into the tree: in particular, the tree resulting from the recursive call has size j (so we had to pay m_j to construct it) with probability $1/n\times(j+1)/(n+1)$, any size from 0 to $n-1$ has the same chance, but we went to that particular side to do the `join` just as if we were inserting something there. And with probability $1/n\times(n-j)/(n+1)$, we will go to the other side, and pay m_{n-1-j}.

On the other hand, if the splitting hyperplane isn't parallel to the discriminating coordinate of the selected root then we will have to `split` the other tree. The probability that L has size j is $1/(n+1)$ and the probability that we choose its root is j/n; if that were the case, we would have to `split` the tree R of size $n-j$. Reversing the roles of L and R in this reasoning, we see that this will contribute a term of the form

$$\begin{aligned}\frac{K-1}{K}\frac{1}{n+1}&\sum_{1\le j\le n}\frac{j}{n}s_{n-j}+\frac{n-j}{n}s_j\\ &=\frac{2(K-1)}{nK}\sum_{0\le j<n}\frac{j+1}{n+1}s_{n-1-j}\\ &=\frac{2(K-1)}{nK}\sum_{0\le j<n}\frac{n-j}{n+1}s_j.\end{aligned}$$

After that, we will have to merge the pieces, for instance, the left subtree of L with R^- and the right subtree of L with R^+, if we had selected the root of L to become the root of the result. Therefore, we have a

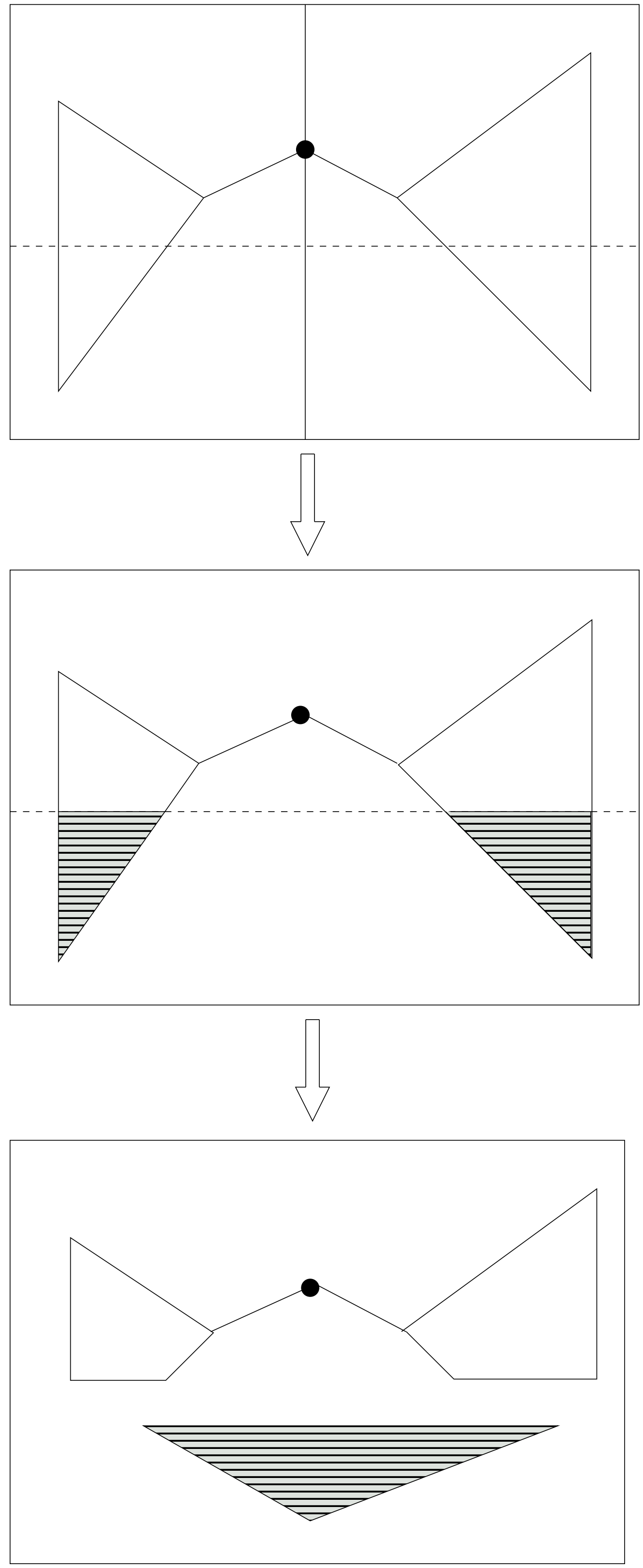

Figure 2: Split algorithm (case 2).

contribution of

$$\frac{K-1}{K}\sum_{0\le j<n}\pi'_{n,j}(m_j+m_{n-1-j}),$$

where $\pi'_{n,j}$ is the probability that the total size of the trees involved in the first `join` is j (then the second join involves $n-1-j$ nodes). Since the two trees that will result from the two `join`'s are random, we know that each has size j with probability $1/n$. Hence, $m_0 = 0$ and for $n > 0$,

(2.2)
$$m_n = 1+\frac{2}{nK}\sum_{0\le j<n}\frac{j+1}{n+1}m_j+\frac{2(K-1)}{nK}\sum_{0\le j<n}m_j + \frac{2(K-1)}{nK}\sum_{0\le j<n}\frac{n-j}{n+1}s_j.$$

Now, it is quite clear that $s_n = m_n$, by simple inspection of the recurrences satisfied by s_n and m_n, and the fact that $s_0 = m_0 = 0$. This result was already proved in [3] and is not surprising at all, given that the two algorithms are so closely intertwined.

Our next section is devoted to the solution of the recurrence (2.1) for `split`, which now reads, for $n > 0$,

(2.3)
$$s_n = 1+\frac{2}{nK}\sum_{0\le j<n}\frac{j+1}{n+1}s_j + \frac{2(K-1)}{nK}\sum_{0\le j<n}s_j+\frac{2(K-1)}{nK}\sum_{0\le j<n}\frac{n-j}{n+1}s_j.$$

3 The expected cost of `split`

We start with a quick and "dirty" solution to the recurrence (2.3) using Roura's master theorem [12]. The shape function $\omega(z)$ for the recurrence (2.3) is

$$\omega(z)=\frac{2}{K}z+\frac{2(K-1)}{K}+\frac{2(K-1)}{K}(1-z).$$

Therefore,

$$\varphi(x)=\int_0^1\omega(z)\,z^x\,dx=\frac{2Kx+6K-4}{K(x+1)(x+2)},$$

and since the toll function is $1 = n^0$, we have that the entropy is $\mathcal{H} = 1-\varphi(0) = -2(K-1)/K$. As the entropy is negative for any $K \ge 2$, the solution to (2.3) is $s_n = m_n = \Theta(n^{\phi(K)})$, where $\phi(K)$ is the positive solution of $\varphi(x) = 1$. In other words, $\phi(K)$ is the positive solution of $Kx^2 + Kx + 4 - 4K = 0$, that is,

$$\phi(K)=\frac{1}{2}\left(\sqrt{17-\frac{16}{K}}-1\right).$$

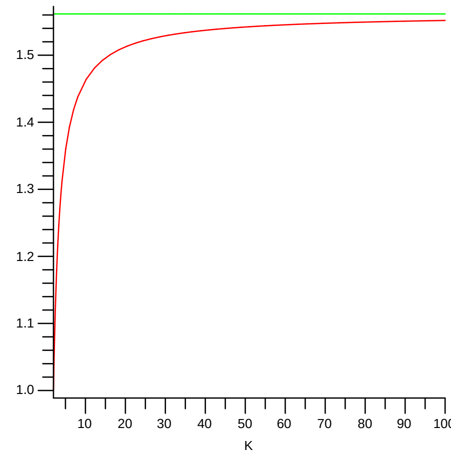

Figure 3: Plot of $\phi(K)$.

Figure 3 depicts the value of $\phi(K)$ and the limiting value $\phi_\infty = \lim_{K\to\infty}\phi(K) = (\sqrt{17}-1)/2 = 1.561552813$. Notice also that $\phi(2) = 1$.

Let us tackle now a different path, which will yield a more precise asymptotic estimate of s_n. Towards this goal, we introduce

$$S(z)=\sum_{n\ge 0}s_n z^n,$$

hence, $s_n = [z^n]S(z)$ is the answer we seek.

Translating the recurrence for s_n into a differential equation for $S(z)$ we obtain the following.

Proposition 3.1. *The generating function $S(z)$ of the expected values s_n satisfies the following second-order linear differential equation:*

(3.4)
$$z\frac{d^2S}{dz^2}+2\frac{1-2z}{1-z}\frac{dS}{dz} - 2\left(\frac{3K-2}{K}-z\right)\frac{S(z)}{(1-z)^2}=\frac{2}{(1-z)^3},$$

with initial conditions $S(0) = 0$ and $S'(0) = 1$.

First of all, it is not hard to find that the function

(3.5)
$$-\frac{1}{2}\left(\frac{K}{K-1}\right)\frac{1}{1-z}$$

is a particular solution of (3.4). Consider now the homogeneous differential equation corresponding to (3.4), that is,

(3.6)
$$zy''(z)+2\frac{1-2z}{1-z}y'(z)-2\left(\frac{3K-2}{K}-z\right)\frac{y(z)}{(1-z)^2}=0.$$

Since there is a singularity at $z = 1$, a natural step is to consider the substitution $y(z) = g(z) \cdot (1-z)^{-\alpha}$. Hence, $r = -\alpha$ satisfies the indicial equation corresponding to (3.6)

$$r(r-1) + 2r - 2\left(\frac{3K-2}{K} - 1\right) = r^2 + r - 4\frac{K-1}{K} = 0, \tag{3.7}$$

whose solutions are

$$r_1 = -\frac{1}{2} - \frac{1}{2}\sqrt{17 - \frac{16}{K}}, \quad r_2 = -\frac{1}{2} + \frac{1}{2}\sqrt{17 - \frac{16}{K}}.$$

Taking derivatives and replacing the proposed form for $y(z)$ with $\alpha = -r_1$ into the differential equation, we obtain the following differential equation for $g(z)$:

$$z(1-z)^2 g''(z) + (2(1-2z) + 2\alpha z)(1-z)g'(z) + \left(z\alpha(\alpha+1) + 2\alpha(1-2z) - 2\left(\frac{3K-2}{K} - z\right)\right)g(z) = 0.$$

Since $-\alpha$ is a root of the indicial equation (3.7) the differential equation above is equivalent to

$$z(1-z)g''(z) + (2 - ((1-\alpha) + (2-\alpha) + 1)z)g'(z) - (1-\alpha)(2-\alpha)g(z) = 0,$$

which is the hypergeometric differential equation (see for instance [1, 10]) with parameters $c = 2$, $a = 1-\alpha$ and $b = 2-\alpha$.

In the particular case when $K = 2$ we fall in the degenerate case of the hypergeometric differential equation because a and b are integers (note that a and b are noninteger for all other $K > 2$). Then the solution of the hypergeometric differential equation is

$$\frac{C_1 + z}{(1-z)^2} + C_2\left(\frac{z^2 - 2z\ln z - 1}{z(1-z)^2}\right)$$

and since the second term of the function does not converge for $z = 0$, we conclude $C_2 = 0$. Finally, taking into account the initial condition $S(0) = 0$ and the particular solution (3.5), the general solution of the differential equation is amazingly simple:

$$S(z) = \frac{z}{(1-z)^2}.$$

Hence, for $K = 2$ we have $s_n = n$.

For $K > 2$, a fundamental system is

$$g(z) = {}_2F_1\left(\left.\begin{matrix}1-\alpha, 2-\alpha\\ 2\end{matrix}\right| z\right) + {}_2F_1\left(\left.\begin{matrix}1-\alpha, 2-\alpha\\ 2\end{matrix}\right| z\right) \cdot \ln z + \sum_{n=1}^{\infty} \frac{a^{\overline{n}} b^{\overline{n}}}{(1+m)^{\overline{n}}} \frac{z^n}{n!}[\psi(a+n) - \psi(a) + \psi(b+n) - \psi(b) - \psi(m+1+n) + \psi(m+1) - \psi(n+1) + \psi(1)] - \sum_{n=1}^{m} \frac{(n-1)!(-m)^{\overline{n}}}{(1-a)^{\overline{n}}(1-b)^{\overline{n}}} z^{-n},$$

where ${}_2F_1$ is the classical hypergeometric function [10], $x^{\overline{n}} = x(x+1)\cdots(x+n-1)$ denotes the nth raising factorial of x, $\psi(z)$ is the logarithimc derivative of the Γ function, and $a = 1-\alpha$ and $b = 2-\alpha$. Furthermore, for the last three terms above we require that $|z| < 1$ and $a, b \neq 0, 1, 2, \ldots, (m-1)$ hold [1]. Since only the first term above converges for $z = 0$, we can conclude that

$$S(z) = C_1 \cdot (1-z)^{-\alpha} \cdot {}_2F_1\left(\left.\begin{matrix}1-\alpha, 2-\alpha\\ 2\end{matrix}\right| z\right) - \frac{1}{2}\frac{1}{1-\rho}\frac{1}{1-z},$$

setting $\alpha = -r_1 = \frac{1}{2}\left(1 + \sqrt{17 - \frac{16}{K}}\right)$ and $\rho = 1/K$. Now, the initial condition $S(0) = 0$ implies $C_1 = \frac{1}{2}\frac{1}{1-\rho}$ and thus we have the following proposition.

Proposition 3.2. *The generating function $S(z)$ of the expected cost of* `split` *is, for any $K > 2$,*

$$S(z) = \frac{1}{2}\frac{1}{1-\rho}\left[(1-z)^{-\alpha} \cdot {}_2F_1\left(\left.\begin{matrix}1-\alpha, 2-\alpha\\ 2\end{matrix}\right| z\right) - \frac{1}{1-z}\right], \tag{3.8}$$

where $\alpha = \alpha(K) = \frac{1}{2}\left(1 + \sqrt{17 - \frac{16}{K}}\right)$, and $\rho = 1/K$.

Extracting the asymptotic estimate for the nth coefficient of $S(z)$ is now routine. Since $S(z)$ has a unique dominant singularity at $z = 1$, applying the standard transfer lemmas of Flajolet and Odlyzko [8] yields

$$s_n \sim \frac{1}{2}\frac{1}{1-\rho}{}_2F_1\left(\left.\begin{matrix}1-\alpha, 2-\alpha\\ 2\end{matrix}\right| 1\right) \cdot \frac{n^{\alpha-1}}{\Gamma(\alpha)}.$$

Finally, as

$${}_2F_1\left(\left.\begin{matrix}1-\alpha, 2-\alpha\\ 2\end{matrix}\right| 1\right) = \frac{\Gamma(2\alpha-1)}{\alpha\Gamma^2(\alpha)},$$

the main result of this section follows.

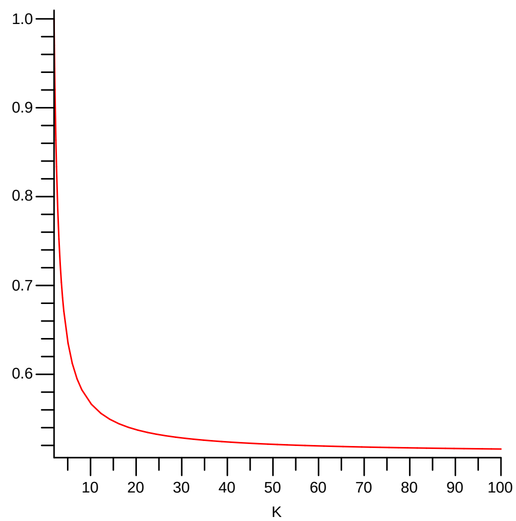

Figure 4: Plot of β as a function of K.

THEOREM 3.1. *The expected cost s_n of splitting a relaxed K-d tree of size n is*

$$s_n = \beta\, n^{\phi(K)} + o(n),$$

with $\rho = 1/K$ and

$$\beta = \frac{1}{2}\frac{1}{1-\rho}\frac{\Gamma(2\alpha - 1)}{\alpha\Gamma^3(\alpha)},$$

$$\phi = \alpha - 1 = \frac{1}{2}\left(\sqrt{17 - \frac{16}{K}} - 1\right).$$

Notice that the theorem above holds even for $K = 2$, as $\phi(2) = 1$ and $\beta(2) = 1$. Observe also that for any $K > 2$, $1 < \phi(K) < (\sqrt{17} - 1)/2 = 1.561552813$. Figure 4 shows a plot of the coefficient β depending on K. It steadily decreases from $\beta(2) = 1$ to $\beta_\infty = \lim_{K\to\infty} \beta(K) = 0.5107413223$.

It is worth noting that the values of s_n that we have just obtained differ from the values for the expected cost $\overline{s}_n$ of splitting a tree of size n given in [3]. This is because we use a different "accounting" scheme, but it can be easily proved that these results are related by

$$\overline{s}_{n+1} = \left(2\frac{K-1}{K} + 1\right) s_n + 1.$$

For instance, for the case $K = 2$, $\overline{s}_{n+1} = 2n + 1 = 2s_n + 1$.

4 The expected cost of insertions and deletions

Last but not least, let us tackle now the expected cost of insertions and deletions. Let I_n be the expected cost of inserting a new node in a relaxed K-d tree of size n. With probability $1/(n + 1)$ we insert the new node at the root of the K-d tree, inducing a split of a tree of size n. With complementary probability the new node must be inserted in either the left or the right subtree depending on the value of the node's key. Since the size of the target subtree is j with probability $1/n$, and the probability of inserting the new item there is then $(j + 1)/(n + 1)$ this yields the recurrence

$$\begin{aligned}\mathbb{E}[I_n] &= \frac{s_n}{n+1} \\ &\quad + \left(1 - \frac{1}{n+1}\right)\left(1 + \frac{2}{n}\sum_{0\le j<n}\frac{j+1}{n+1}\,\mathbb{E}[I_j]\right) \\ &= \frac{s_n}{n+1} + 1 + \mathcal{O}\left(\frac{1}{n}\right) + \frac{2}{n+1}\sum_{0\le j<n}\frac{j+1}{n+1}\,\mathbb{E}[I_j]\,.\end{aligned}$$

The solution of the previous recurrence is

$$\mathbb{E}[I_n] = \frac{2}{\mathcal{H}'}\ln n + \mathcal{O}(1) = 4\ln n + \mathcal{O}(1),$$

for $K = 2$, with $\mathcal{H}' = 1/2$. For $K > 2$,

$$\begin{aligned}\mathbb{E}[I_n] &= \frac{\beta}{1 - \frac{2}{\phi+1}} n^{\phi-1} + 2\ln n + \mathcal{O}(1) \\ &= \beta\frac{\phi+1}{\phi-1} n^{\phi-1} + 2\ln n + \mathcal{O}(1).\end{aligned}$$

where β and ϕ are as in Theorem 3.1. These results can be readily obtained applying Roura's master theorem [12]. Figure 5 shows a plot of $B = \beta(\phi + 1)/(\phi - 1)$. The values of B range from its maximum value $B(3) = 8.075320731\ldots$ to its minimum value $B_\infty = \lim_{K\to\infty} B(K) = 2.329773514\ldots$.

Last but not least, the expected cost of random deletions $\mathbb{E}[D_n]$ is of the same order of magnitude as for insertions. The recurrence satisfied by $\mathbb{E}[D_n]$ is very similar to the one above:

$$\mathbb{E}[D_n] = \frac{m_{n-1}}{n} + \left(1 - \frac{1}{n}\right)\left(1 + \frac{2}{n}\sum_{0\le j<n}\frac{j}{n}\,\mathbb{E}[D_j]\right).$$

With probability $1/n$, the current node is the one to be deleted with cost m_{n-1}, as the result of the `join` of its two subtrees will have total size $n-1$. With probability $1 - 1/n$, the item to be deleted will be in one of the subtrees. Since

$$\frac{m_{n-1}}{n} = \frac{s_n}{n+1}\left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right)$$

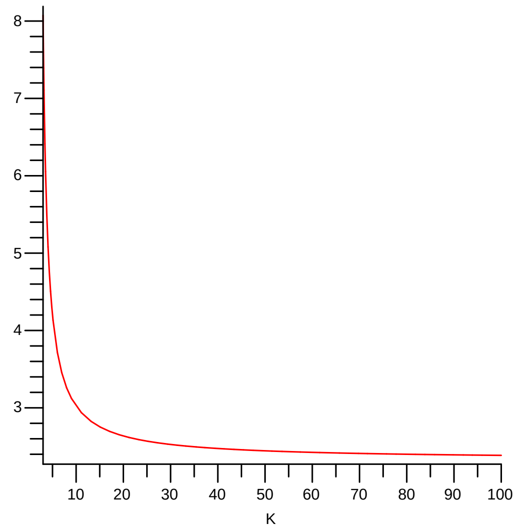

Figure 5: Plot of B as a function of K.

and the shape function for both recurrences is the same, we conclude that

$$\begin{aligned}\mathbb{E}[D_n] &= \mathbb{E}[I_n]\left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) \\ &= B \cdot n^{\phi-1} + 2\ln n + \mathcal{O}(1),\end{aligned}$$

if $K > 2$, and $\mathbb{E}[D_n] = 4\ln n + \mathcal{O}(1)$, if $K = 2$.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* Dover, 1964.

[2] J. L. Bentley. Multidimensional binary search trees used for associative retrieval. *Communications of the ACM*, 18(9):509–517, 1975.

[3] N. Broutin, K. Dalal, L. Devroye, and E. McLeish. The k-d treap. *ACM Trans. on Algorithms*, 2006. Accepted for publication.

[4] P. Chanzy, L. Devroye, and C. Zamora-Cura. Analysis of range search for random k-d trees. *Acta Informatica*, 37:355–383, 2001.

[5] L. Devroye, J. Jabbour, and C. Zamora-Cura. Squarish k-d trees. *SIAM Journal on Computing*, 30:1678–1700, 2000.

[6] A. Duch, V. Estivill-Castro, and C. Martínez. Randomized K-dimensional binary search trees. In K.-Y. Chwa and O. H. Ibarra, editors, *Int. Symposium on Algorithms and Computation (ISAAC'98)*, volume 1533 of *LNCS*, pages 199–208. Springer-Verlag, 1998.

[7] A. Duch and C. Martínez. On the average performance of orthogonal range search in multidimensional data structures. *Journal of Algorithms*, 44(1):226–245, 2002.

[8] Ph. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM Journal on discrete Mathematics*, 3(1):216–240, 1990.

[9] Ph. Flajolet and C. Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33(2):371–407, 1986.

[10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics, 2nd edition.* Addison–Wesley, 1994.

[11] C. Martínez, A. Panholzer, and H. Prodinger. Partial match queries in relaxed multidimensional search trees. *Algorithmica*, 29(1–2):181–204, 2001.

[12] S. Roura. Improved master theorems for divide-and-conquer recurrences. *Journal of the ACM*, 48(2):170–205, 2001.