

Hindawi Publishing Corporation
Modelling and Simulation in Engineering
Volume 2014, Article ID 748941, 5 pages
<http://dx.doi.org/10.1155/2014/748941>



Research Article

On Counting and Embedding a Subclass of Height-Balanced Trees

Indhumathi Raman

School of Information Technology and Engineering, VIT University, Vellore 632014, India

Correspondence should be addressed to Indhumathi Raman; indhumathi.raman@gmail.com

Received 7 February 2014; Revised 5 May 2014; Accepted 5 May 2014; Published 26 May 2014

Academic Editor: Aiguo Song

Copyright © 2014 Indhumathi Raman. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A height-balanced tree is a rooted binary tree in which, for every vertex v , the difference in the heights of the subtrees rooted at the left and right child of v (called the balance factor of v) is at most one. In this paper, we consider height-balanced trees in which the balance factor of every vertex beyond a level is 0. We prove that there are 2^{2^l-1} such trees and embed them into a *generalized join of hypercubes*.

1. Introduction

The problem of efficiently implementing parallel algorithms on parallel computers has been studied as a graph embedding problem, which is to embed the communication graph G_A (underlying a parallel algorithm A) within the network topology H_N of a parallel computer N with minimum communication overhead. The working of the parallel algorithm A when implemented on N can be described by embedding G_A into H_N . Mathematically, an embedding ϕ of a (guest) graph $G(V, E)$ into a (host) graph $H(W, F)$ is a one-one function $\phi : V \rightarrow W$ such that if (u, v) is an edge in G , then $\phi(u)$ and $\phi(v)$ are connected by a path in H . We use $\phi : V \rightarrow W$ and $\phi : G \rightarrow H$ analogously. The following are two of the parameters that measure the quality of an embedding.

- (i) $\text{Dilation}(\phi) := \max\{\text{dist}_H(\phi(u), \phi(v)) : (u, v) \in E\}$.

By $\text{dist}_H(x, y)$ we mean the shortest distance between x and y in graph H . The dilation of an embedding measures the maximum time required to route a message in the host graph, between any two vertices. If $\text{dilation}(\phi) = 1$, then G is isomorphic to a subgraph of H and we write $G \subseteq H$. If $\text{dilation}(\phi) = 2$, then in this paper we denote it by $G \hookrightarrow H$.

- (ii) $\text{Expansion}(\phi) := |W|/|V|$. This parameter measures the relative number of unutilized processors

in the host graph. Since ϕ is a one-one function, $\text{expansion}(\phi)$ is at least 1.

A goal of a parallel algorithm designer is to map (or embed) the algorithm graph G into an interconnection network H with dilation and expansion kept as minimum as possible. However, for most embedding problems, it is very difficult to obtain an embedding that minimizes these parameters simultaneously. Therefore, some tradeoffs among the parameters must be made; see Leighton [1].

Trees generally form the underlying data structure for several parallel algorithms that employ divide-and-conquer rule, branch-and-bound technique, and so on. A self-balancing binary search tree that ensures a logarithmic running time (in both average and worst case) on operations such as insertion, deletion, and rotation, regardless of the order of the data inserted, is a height-balanced tree. Insertions and deletions may require the tree to be rebalanced by one or more tree rotations. However, these rotations take constant time. The height-balanced trees are also called AVL trees, named after their inventors Adelson-Velskii and Landis [2], and their computational aspects are discussed in [3–5]. The computational efficiency of height-balanced data structures has motivated a vast study in parallel computations too. They have been used for concurrent search, concurrent insertion, and information storage and retrieval and implement parallel



FIGURE 1: Examples of height-balanced trees.

dictionary routines; see [4–7]. A formal definition of height-balanced trees is given in the next section.

On the other hand, among the interconnection networks of parallel computers, the binary *hypercube* has received much attention. An n -dimension *hypercube* (denoted by Q_n) has 2^n vertices each labelled with a binary string of length n and two vertices are adjacent if and only if their labels differ in exactly one position. The hypercube topology has several attractive features. Its popularity is due to the fact that the nodes of a hypercube are symmetrically placed and the number of interprocessor connections per node is small. Another important property of a hypercube is its ability to efficiently simulate the message routings of other interconnection networks. An extensive survey of embedding various graphs into hypercubes is given in [1, 8–10].

In this paper, we consider a subclass $\text{HB}(h, t)$ of height-balanced trees and we analyse the size of the subclass by counting the number of trees in it. We also prove that every height-balanced tree $T_h \in \text{HB}(h, t)$ is embeddable in a generalized join of hypercubes with dilation 2.

2. Preliminaries

Definition 1. A rooted binary tree T is said to be height-balanced if, for every vertex $v \in V(T)$, the heights of the subtrees, rooted at the left and right child of v , differ by at most one.

A classical example of a height-balanced tree is the complete binary tree. Figure 1 shows a few small height-balanced trees.

In this paper, without loss of generality, we assume that the height of the left subtree is not less than the height of the right subtree for every $v \in V(T)$. With any height-balanced tree T , we associate a function $\mathbf{b}_T : V(T) \rightarrow \{0, 1\}$ defined by $\mathbf{b}_T(v) :=$ the difference between the heights of left and right subtrees of v , for every vertex $v \in V(T)$. This difference $\mathbf{b}_T(v)$ is called the balance factor of v . The balance condition, being binary, ensures that the height of a height-balanced tree is logarithmic of its size; refer to [11].

It is usually assumed that the height of an empty tree is -1 and that of a single-vertex tree is 0. Consequently, for a leaf $l \in V(T)$, $\mathbf{b}_T(l) = 0$. Clearly,

$$\begin{aligned} \mathbf{b}_T(v) = 0, \quad & \text{for every } v \in V(T) \\ & \text{iff } T \text{ is a complete binary tree.} \end{aligned} \quad (1)$$

If T_h is a height-balanced tree of height h , we denote its levels by L_0, L_1, \dots, L_h , where $L_i = \{v \in V(T_h) : \text{dist}(v, \text{root}) = i\}$, $0 \leq i \leq h$. It is shown in [2] that the

$$\begin{aligned} & \text{subtree of } T_h \text{ induced by vertices} \\ & \text{in } L_0 \cup L_1 \cdots \cup L_{\lfloor h/2 \rfloor} \text{ is complete.} \end{aligned} \quad (2)$$

Let $\text{HB}(h, t)$ (where $0 \leq t \leq h$) denote the subclass of height-balanced trees T with height h such that

$$\mathbf{b}_T(v) = \begin{cases} 0 & \text{if } v \in L_t \cup L_{t+1} \cup \cdots \cup L_h \\ 0 \text{ or } 1 & \text{if } v \in \bigcup_{L_t} \cup L_{t+1} \cup \cdots \cup L_h. \end{cases} \quad (3)$$

So $\text{HB}(h, 0) \subset \text{HB}(h, 1) \subset \cdots \subset \text{HB}(h, h)$. While $\text{HB}(h, 0)$ contains only one element, namely, the complete binary tree of height h , $\text{HB}(h, h)$ is the class of all height-balanced trees of height h . Let $v(t, j)$ be the j th vertex from left at level t ; then it follows from (1) and from the definition of $\text{HB}(h, t)$ that

$$\begin{aligned} & \text{the subtree of } T \text{ rooted at } v(t, j) \in L_t \\ & \text{is a complete binary tree.} \end{aligned} \quad (4)$$

By (2) and (4), the number of vertices in $T \in \text{HB}(h, t)$, for $0 \leq t \leq \lfloor h/2 \rfloor$, is

$$|V(T)| = (2^t - 1) + \sum_{v(t,j) \in L_t} (2^{h_j+1} - 1) = \sum_{j=1}^{2^t} (2^{h_j+1}) - 1. \quad (5)$$

3. Counting Trees of $\text{HB}(h, t)$

Theorem 2. *The class $\text{HB}(h, t)$ of height-balanced trees contains exactly 2^{2^t-1} trees.*

Proof. The idea of the proof is to establish a one-to-one correspondence between $\text{HB}(h, t)$ and $\mathbf{Z}_2^{2^t-1}$. That is, for every $T \in \text{HB}(h, t)$, we will prove that there exists a sequence $(x_1, x_2, \dots, x_{2^t-1})$, $x_i \in \{0, 1\}$, $1 \leq i \leq 2^t - 1$, such that $\mathbf{b}_T(v(j, k)) = x_{2^j+k-1}$, $0 \leq j \leq t-1$, $1 \leq k \leq 2^j$, and vice versa.

We first assume the existence of a tree $T \in \text{HB}(h, t)$ and associate an element of $\mathbf{Z}_2^{2^t-1}$ to T . Given the tree T , we order the vertices of $L_0 \cup L_1 \cup \cdots \cup L_{t-1}$ according to the breadth first search (BFS), say, $(v_1, v_2, \dots, v_{2^t-1})$. For $1 \leq i \leq 2^t - 1$, let $x_i = \mathbf{b}_T(v_i)$. Then the sequence $(x_1, x_2, \dots, x_{2^t-1})$ is the required sequence.

We next prove the converse. For a given sequence $(x_1, x_2, \dots, x_{2^t-1})$ of 0's and 1's, we construct a tree T (Figure 2) such that (1) $T \in \text{HB}(h, t)$ and (2) $\mathbf{b}_T(v(t, j)) = x_{2^t-j+1}$ as follows. Consider a complete binary tree C_t of height t . Let l_j denote the j th leaf (where $1 \leq j \leq 2^t$) from left in level t . For every j , $1 \leq j \leq 2^t$, we root a complete binary tree C_{h_j} to l_j (i.e., l_j becomes the root of C_{h_j}). The height h_j of C_{h_j} is calculated as follows:

$$h_j = \begin{cases} h - t, & \text{if } j = 1 \\ h_j - 2^{s_j} - x_{\lfloor 2^{t-s_j-1} + \lfloor (j-1)/2^{s_j+1} \rfloor - 1 \rfloor}, & \end{cases} \quad (6)$$

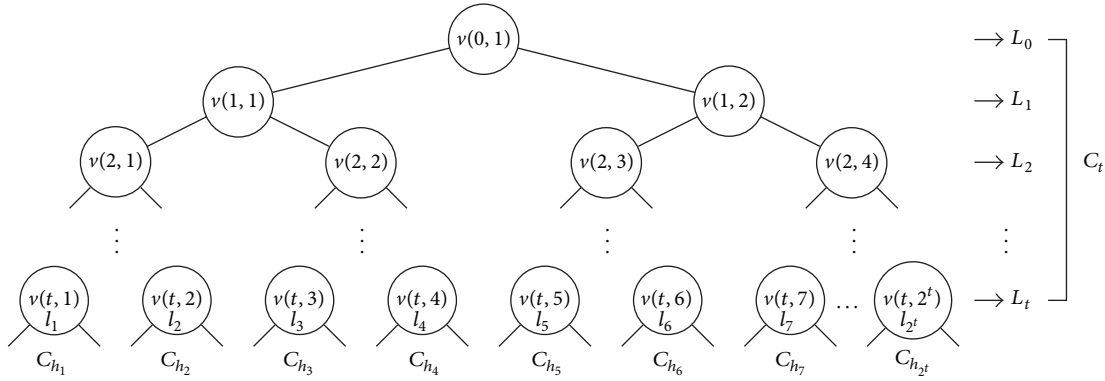


FIGURE 2: $T \in \text{HB}(h, t)$. The circles represent the vertices of the induced CBT from level 0 to t . The text inside the circle $v(i, j)$ for $0 \leq i \leq t$, $1 \leq j \leq 2^i$ denotes the j th vertex from left at L_i . The vertex $v(t, j)$ is also denoted by l_j .

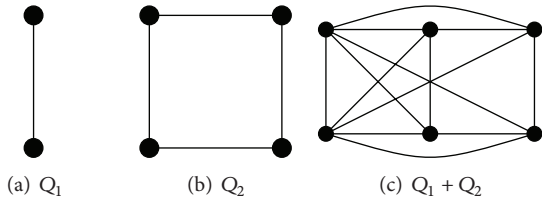


FIGURE 3: The graph join $Q_1 + Q_2$.

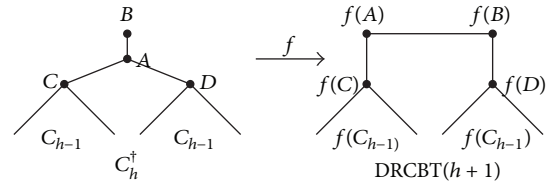


FIGURE 4: $C_h^+ \leftrightarrow \text{DRCBT}(h+1)$.

where

$$s_j = \begin{cases} 0, & \text{if } j \text{ is even} \\ \text{greatest power of 2 that divides } j-1, & \text{if } j \text{ is odd.} \end{cases} \quad (7)$$

Claim 1 ($T \in \text{HB}(h, t)$). From (6), we note that $h_j \leq h - t$. Therefore, the height of T is height of $C_t + \max_{1 \leq j \leq 2^t} h_j = t + (h - t) = h$. In T , for all $i \geq t$, every vertex $u \in L_i$ is a vertex of some C_{h_j} , $1 \leq j \leq 2^t$. Hence, $\mathbf{b}_T(u) = 0$ for all $u \in (L_t \cup L_{t+1} \cup \dots \cup L_h)$. This implies that $T \in \text{HB}(h, t)$.

Claim 2 (for $0 \leq j \leq t-1$ and $1 \leq k \leq 2^j$, $\mathbf{b}_T(v(j, k)) = x_{2^{j+k-1}}$). Let $T_{j,k}$ denote the subtree rooted at the vertex $v(j, k)$ of T . Hence by definition, the balance factor $\mathbf{b}_T(v(j, k))$ of the vertex $v(j, k)$ is the difference between the heights of the left subtree $T_{j+1, 2k-1}$ and the right subtree $T_{j+1, 2k}$ of $v(j, k)$. That is,

$$\begin{aligned} \mathbf{b}_T(v(j, k)) &:= \text{Height of } T_{j+1, 2k-1} - \text{Height of } T_{j+1, 2k} \\ &= [(t-j-1) + \text{Height of } T_{t, k2^{t-j}-2^{t-j+1}}] \\ &\quad - [(t-j-1) + \text{Height of } T_{t, k2^{t-j}-2^{t-j+1}+1}]. \end{aligned} \quad (8)$$

Let m_1 and m_2 denote the integers $k2^{t-j} - 2^{t-j} + 1$ and $k2^{t-j} - 2^{t-j-1} + 1$, respectively. Since the subtree rooted at a vertex in level t is a complete binary tree, we have $T_{t, m_1} \simeq C_{h_{m_1}}$ and $T_{t, m_2} \simeq C_{h_{m_2}}$ and $\mathbf{b}_T(v(j, k)) = h_{m_1} - h_{m_2}$. In order to calculate the difference $h_{m_1} - h_{m_2}$, we consider two cases.

Case 1 ($0 \leq j < t-1$). On rewriting the values of m_1 and m_2 we have $m_1 = 2^{t-j}(k-1) + 1$ and $m_2 = 2^{t-j}(k-1) + 2^{t-j-1} + 1$. We note that the greatest powers of 2 that divides $m_1 - 1$ and $m_2 - 1$ are $t-j$ and $t-j-1$. Hence, by following (6), we have $s_{m_1} = t-j$ and $s_{m_2} = t-j-1$, $h_{m_1} = h_{m_1 - 2^{t-j} - x_{[2^{t-j} + (m_1 - 1)/2^{t-j} - 1]}}$, and then,

$$\begin{aligned} h_{m_2} &= h_{m_2 - 2^{t-j-1} - x_{[2^{t-j-1} + (m_2 - 1)/2^{t-j-1} - 1]}} \\ &= h_{m_1} - x_{[2^{j+k-1} + (k2^{t-j} - 2^{t-j-1})/2^{t-j} - 1]} \\ &= h_{m_1} - x_{2^{j+k-1}}. \end{aligned} \quad (9)$$

Case 2 ($j = t-1$). In this case, $m_1 = 2k-1$ and $m_2 = 2k$. Since m_2 is even, $s_{m_2} = 0$ and

$$\begin{aligned} h_{m_2} &= h_{m_2-1} - x_{[2^{t-1} + (m_2-1)/2 - 1]} \\ &= h_{m_1} - x_{2^{j+k-1}}. \end{aligned} \quad (10)$$

Hence in both cases, $\mathbf{b}_T(v(j, k)) = h_{m_1} - h_{m_2} = x_{2^{j+k-1}}$. We have proved the claims and hence the theorem. \square

4. Embedding Trees of $\text{HB}(h, t)$

In this section, we embed every tree $T \in \text{HB}(h, t)$ into a generalized join of hypercubes. Before we proceed to the result, we define the join of graphs and prove certain preliminary lemmas.

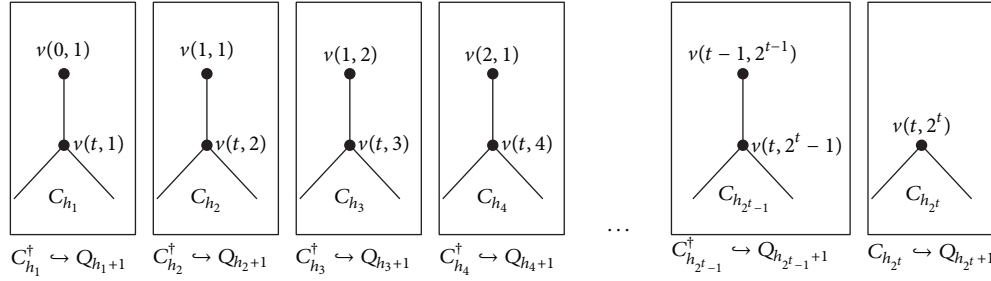


FIGURE 5: Embedding of $T \in \text{HB}(h, t)$ in $\sum_{j=1}^{2^t} Q_{h_{j+1}}$. The intermediate edges between every pair of vertices are not shown for clarity purpose.

Definition 3. The join (denoted by $G_1 + G_2$) of two graphs G_1 and G_2 is defined as follows:

$$\begin{aligned} V(G_1 + G_2) &= V(G_1) \cup V(G_2) \\ E(G_1 + G_2) &= E(G_1) \cup E(G_2) \\ &\cup \{(u, v) : u \in V(G_1), v \in V(G_2)\}, \end{aligned} \quad (11)$$

that is, joining each vertex of G to every vertex of H . Figure 3 shows the join $Q_1 + Q_2$ of the hypercubes Q_1 and Q_2 .

Hence from the definition of graph join we have $|V(G_1 + G_2)| = |V(G_1)| + |V(G_2)|$ and $|E(G_1 + G_2)| = |E(G_1)| + |E(G_2)| + |V(G_1)||V(G_2)|$. We denote the generalized join $(G_1 + (G_2 + \dots + (G_{n-1} + G_n)))$ of graphs G_1, G_2, \dots, G_n by $\sum_{i=1}^n G_i$. We note that $|V(\sum_{i=1}^n G_i)| = \sum_{i=1}^n |V(G_i)|$.

Lemma 4. Let C_h^\dagger be a supertree of the complete binary tree C_h of height h with root A formed by adding a new vertex B and the edge (A, B) . Let $\text{DRCBT}(h+1)$ be the double rooted complete binary tree on 2^{h+1} vertices. Then $C_h^\dagger \hookrightarrow \text{DRCBT}(h+1)$.

Proof. Figure 4 shows the tree C_h^\dagger . We call B as the *deep root* of C_h^\dagger . Consider an embedding $f: V(C_h^\dagger) \rightarrow V(\text{DRCBT}(h+1))$ as shown in Figure 4. Such an embedding f has dilation 2 since the edge $(A, D) \in E(C_h^\dagger)$ is mapped onto a path of length 2 in $\text{DRCBT}(h+1)$ and this is the maximum shortest distance. The embedding f has expansion 1 since both C_h^\dagger and $\text{DRCBT}(h+1)$ have 2^{h+1} vertices. Hence, in short, $C_h^\dagger \hookrightarrow \text{DRCBT}(h+1)$. \square

It is known that $\text{DRCBT}(h+1)$ is a subgraph of Q_{h+1} ; see Nebeský [12]. Hence, this fact along with Lemma 4 implies the following.

Lemma 5. One has $C_h^\dagger \hookrightarrow Q_{h+1}$.

In the next theorem, we give an embedding of every tree of $\text{HB}(h, t)$ into a generalized join of hypercubes.

Theorem 6. Let $T \in \text{HB}(h, t)$ and let h_j be the height of the complete binary tree rooted at the j th vertex from left at a level t ($0 \leq t \leq \lfloor h/2 \rfloor$) of T . Then $T \hookrightarrow \sum_{j=1}^{2^t} Q_{h_{j+1}}$.

Proof. Consider $T \in \text{HB}(h, t)$. Let $v(i, j)$ denote the i th vertex from left in level j . Decompose the tree into a forest of trees by deleting the set of edges $\{(v(t-1, k), v(t, 2k-1)), (v(t-1, k), v(t, 2k))\}$ for $1 \leq k \leq 2^{t-1}$. Such a forest consists of complete binary trees C_{h_j} rooted at $v(t, j)$ (for $1 \leq j \leq 2^t$) and a complete binary tree C_{t-1} induced by the vertices of T from levels 0 to $t-1$. For $1 \leq j \leq 2^t - 1$, let $v(a, b)$ where $2^a + b - 1 = j$ be the deep root of C_{h_j} . Consider the embeddings $f_j: C_{h_j}^\dagger \hookrightarrow Q_{h_{j+1}}$ (such embeddings exist by Lemma 5). For $j = 2^t$, consider $f_j: C_{h_j} \hookrightarrow Q_{h_{j+1}}$ (such an embedding exists by [12]). The vertices $v(i, j)$ for $0 \leq i \leq t$ and $1 \leq j \leq 2^t$ form a complete graph (a graph in which every pair of distinct vertices is connected by a unique edge) in $\sum_{j=1}^{2^t-1} Q_{h_{j+1}}$ and hence contain as subgraph the deleted set of edges of T . Such an embedding technique is depicted in Figure 5. The dilation of the embedding is clearly 2 and hence $T \hookrightarrow \sum_{j=1}^{2^t} Q_{h_{j+1}}$. The number of vertices of $\sum_{j=1}^{2^t} Q_{h_{j+1}}$ is $\sum_{j=1}^{2^t} 2^{h_{j+1}}$ and by (5), we have $|V(T)| = \sum_{j=1}^{2^t} (2^{h_{j+1}}) - 1$. Hence the expansion of the embedding is almost 1 and is optimal. \square

5. Conclusion

In this paper, we have considered a subclass $\text{HB}(h, t)$ of height-balanced trees and have analysed the size of this subclass. We have also embedded every tree in this subclass into a generalized join of hypercubes.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

- [1] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, Calif, USA, 1992.
- [2] G. M. Adelson-Velskii and E. M. Landis, "An algorithm for the organization of information," *Soviet Mathematics Doklady*, vol. 3, pp. 1259–1262, 1962.
- [3] P. Crescenzi and A. Piperno, "Optimal-area drawings of AVL trees," in *Proceedings of the DIMACS International Workshop on*

- Graph Drawings*, vol. 894 of *Lecture Notes in Computer Science*, pp. 307–317, 1994.
- [4] C. S. Ellis, “Concurrent search and insertion in AVL trees,” *IEEE Transactions on Computers*, vol. 29, no. 9, pp. 811–817, 1980.
- [5] M. Medidi and N. Deo, “Parallel dictionaries using AVL trees,” *Journal of Parallel and Distributed Computing*, vol. 49, no. 1, pp. 146–155, 1998.
- [6] C. C. Foster, “Information storage and retrieval using AVL trees,” in *Proceedings of the ACM 20th National Conference*, pp. 192–205, 1965.
- [7] P. L. Karlton, S. H. Fuller, R. E. Scroggs, and E. B. Kaehler, “Performance of height-balanced trees,” *Communications of the ACM*, vol. 19, no. 1, pp. 23–28, 1976.
- [8] S. N. Bhatt, F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg, “Efficient embeddings of trees in hypercubes,” *SIAM Journal on Computing*, vol. 21, no. 1, pp. 161–162, 1992.
- [9] S. A. Choudum and S. Lavanya, “Embedding a subclass of trees into hypercubes,” *Discrete Mathematics*, vol. 311, no. 10–11, pp. 866–871, 2011.
- [10] I. Havel, “On Hamiltonian circuits and spanning trees of hypercubes,” *Časopis pro Pěstování Matematiky*, vol. 109, no. 2, pp. 135–152, 1984.
- [11] S. A. Choudum and I. Raman, “Embedding height balanced trees and Fibonacci trees in hypercubes,” *Journal of Applied Mathematics and Computing*, vol. 30, no. 1–2, pp. 39–52, 2009.
- [12] L. Nebeský, “On cubes and dichotomic trees,” *Časopis pro Pěstování Matematiky*, vol. 99, pp. 164–167, 1974.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

