Linear Layouts of Generalized Hypercubes

Koji Nakano

Advanced Research Laboratory, Hitachi, Ltd., Hatoyama, Saitama 350-03, Japan
e-mail : nakano@harl.hitachi.co.jp

Abstract. This paper studies linear layouts of generalized hypercubes, a d-dimensional c-ary clique and a d-dimensional c-ary array, and evaluates the bisection width, cut width, and total edge length of them, which are important parameters to measure the complexity of them in terms of a linear layout.

1 Introduction

This paper treats two kinds of generalized hypercubes: *a d-dimensional c-ary clique* (abbreviated as Cc_d) and *a d-dimensional c-ary array* (abbreviated as Ac_d). Cc_d has nodes labeled by the c^d integers from 0 to $c^d - 1$. The nodes are connected by the edges if and only if the c-ary representations of their labels differ by one and only one digit (Fig. 1). *An n-node c-ary clique* (abbreviated as $Cc_{(n)}$), which is a more generalized graph, has n nodes labeled by the integers from 0 to $n-1$ and connected in the same way as Cc_d. Note that n is not restricted to a power of c. Ac_d has the same nodes as Cc_d. The nodes are connected if and only if the c-ary representations of their labels differ by one and only one digit and the absolute value of the difference in that digit is 1 (Fig. 2).

Several algorithms on parallel computers based on Cc_d and Ac_d topologies have been shown [1, 7]. It is very important to analyze topological properties of them, because they are very attractive as network topologies of future parallel computers. Furthermore, Cc_d and Ac_d include typical topologies which are used for parallel machines: Cc_1 corresponds to a c-node clique (or a complete graph), Ac_1 corresponds to a c-node linear array, Ac_2 corresponds to a $c \times c$-node 2-dimensional array, Ac_3 corresponds to a $c \times c \times c$-node 3-dimensional array, and both $C2_d$ and $A2_d$ correspond to a d-dimensional (binary) hypercube. Therefore, the results presented in this paper can be applied to these topologies.

A linear layout of a graph $G = (V, E)$ (where V and E are a set of nodes and a set of edges, respectively) is a one-to-one mapping $L : V \rightarrow \{0, 1, 2, \ldots, |V| - 1\}$. This means that each u ($\in V$) is assigned to the position $L(u)$ on the baseline. Examples of linear layouts of $C4_2$ and $A4_2$ are illustrated in Figs. 3 and 4, where each node u is assigned to the position $L(u)$, that is, $L(u) = u$ for all u. We call such the layout L *the label order layout*. Note that a linear layout can take any permutation (i.e. $|V|!$ permutations), not just the label order layout.

The complexity of $G = (V, E)$ in terms of a linear layout is measured by the following parameters: *the (minimum) bisection width*, *the cut width*, and *the total edge length*. These parameters are defined as follows. *The cut of a*

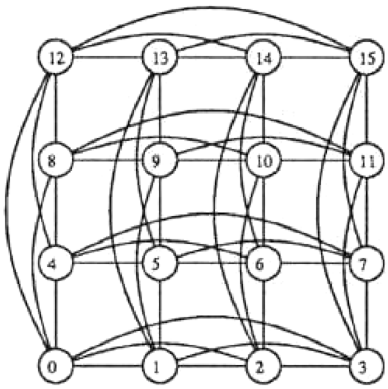

Fig. 1. A 2-dimensional 4-ary array

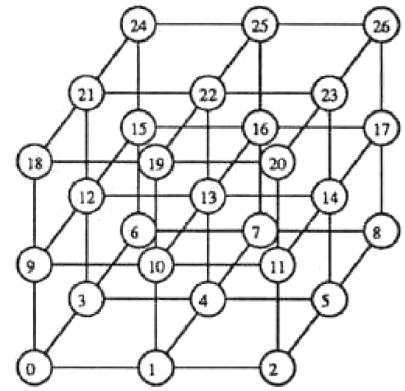

Fig. 2. A 3-dimensional 3-ary array

graph G under a linear layout L at a gap i is a set of edges connecting a node at a position less than i and one at a position larger than or equal to i, i.e. $C(G, L, i) \stackrel{\text{def}}{=} \{(u, v) \in E | 0 \leq L(u) < i \leq L(v) \leq |V| - 1\}$. *The bisection width of a graph G* is the minimum number of edges in $C(G, L, \lfloor |V|/2 \rfloor)$ over all linear layouts, i.e. $\min_L |C(G, L, \lfloor |V|/2 \rfloor)|$. In other words, the bisection width of a graph is the minimum number of edges which must be removed to separate the graph into two disjoint and equal-sized subgraphs. *The cut width of a graph G under a linear layout L* is the maximum of $|C(G, L, i)|$ over all gaps i, i.e. $\max_i |C(G, L, i)|$. *The cut width of a graph G* is the minimum cut width over all linear layouts, i.e. $\min_L \max_i |C(G, L, i)|$. This parameter indicates the number of tracks required by the best linear layout. We will define that *the length of edge $(u, v) \in E$ under a linear layout L* is $|L(u) - L(v)|$. Then, *the total edge length of a graph G under a linear layout L* is $\sum_{(u,v) \in E} |L(u) - L(v)|$. Furthermore, *the total edge length of a graph G* is defined as the minimum of this value over all linear layouts, i.e. $\min_L \sum_{(u,v) \in E} |L(u) - L(v)|$. Obviously, the total edge length is equal to the total cut, i.e. $\min_L \sum_{i=1}^{|V|-1} |C(G, L, i)|$

It is very important to compute exact values of them, because they determine the lower bound of the layout area in the VLSI model. For example, the layout area of a processor network is at least $\Omega(B^2)$ if the corresponding graph has bisection width B [6, 13], and the number of tracks of a processor network in a horizontal layouts requires C layers if the corresponding graph has cut width C. The total edge length has applications to the coding theory [5] and storage management [12]: Minimizing the total edge length of generalized hypercubes corresponds to minimizing the error of a c-ary channel, and to minimizing the efficiency of managing a d-dimensional data structure in a paging environment. However, the problem to compute the exact values of them are hard problem: For a given graph and an integer k, the problem to determine whether the

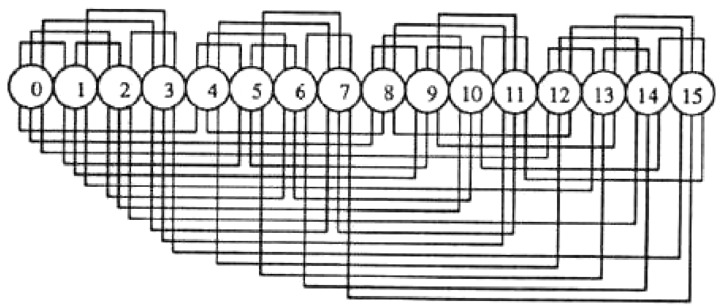

Fig. 3. The label order layout of a 2-dimensional 4-ary clique

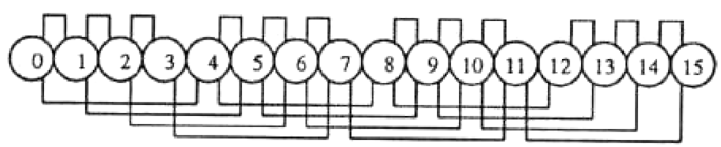

Fig. 4. The label order layout of a 2-dimensional 4-ary clique

bisection width of the graph is at most k is NP-complete [4]. Similarly, the problem to determine the cut width is NP-complete even if the degree of the graph is restricted [8].

Several articles have been devoted to the evaluation of them. Brebner [2], Manabe et al. [9], and Nakano et al. [10] have proved that the bisection width of a d-dimensional binary hypercube is 2^{d-1} using different methods. Leighton [7] showed that the bisection width of Ac_d is c^{d-1} if c is even by embedding a directed complete graph in Ac_d. Wada et al. [14] proved that the bisection width of Cc_d is $c^{d+1}/4$ if c is even in a similar way to the Leighton's proof. However, they did not get the exact value of it when c is odd: the bisection width of Cc_d takes a value between $\lceil c^{d+1}/4 - 1/(4c^{d-1})\rceil$ and $(c+1)(c^d-1)/4$ (inclusive). Nakano et al. [10] also proved that the cut width of $C2_d$ is $\lfloor 2^{d+1}/3 \rfloor$. Wada et al. [15] also proved that the cut width of Cc_d is at most $c^2(c^d-1)/\{4(c-1)\}$. Niepel et al. [11] showed that the total edge length of an $n \times 2$-node array is $5n-4$ and conjectured that that of an $n \times m$-node array is $n(m^2+m-1)-m^2$. Harper [5] showed that the total edge length of a d-dimensional hypercube is $2^{d-1}(2^d-1)$. DeMillo et al. [3] showed that the total edge length of 2-dimensional hypercube is at least $n^3/6$.

In this paper, we will evaluate the bisection width, cut width, and total edge length of Cc_d and of Ac_d. In Section 2, we consider how many edges a subgraph of $Cc_{(m)}$ with n ($n \leq m$) nodes may have, and show that $Cc_{(n)}$ has the largest number of edges of all subgraphs with n nodes. In other words, $Cc_{(n)}$ is the maximum subgraph of $Cc_{(m)}$ if $n \leq m$. Section 3 uses this fact to get the exact values of the bisection width, cut width, and total edge length of Cc_d. Section 4 presents the method for converting Cc_d into Ac_d and get exact value of the bisection width of Ac_d, and nearly exact values of the cut width and the total edge length of Ac_d. See Table 1 for comparing our results and previously known results.

Table 1. Our results and previously known results

	Bisection width	Cut width	Total edge length
hypercube	Brebner [2] Manabe [9] Nakano [10] exact	Nakano [10] exact	Harper [5] exact
d-dimensional c-ary clique	Wada [14] exact when c is even	Wada [15] only upper bound	
	This paper exact	This paper exact	This paper exact
d-dimensional c-ary array	Leighton [7] exact when c is even		DeMillo [3] lower bound when $d = 2$§
	This paper exact	This paper nearly exact†	This paper nearly exact‡

†The upper bound is about $1 + 2/\{(c+2)(c-1)\}$ times as large as the lower bound.
‡The upper bound is about $3c/\{2(c+1)\}$ times as large as the lower bound.
§DeMillo's lower bound is $c^3/6$, while that of us is about $2c^3/3$.

2 Maximum subgraph of Cc_d

The main result of this paper is due to the following theorem:

Theorem 1. *$Cc_{(n)}$ is a maximum subgraph of $Cc_{(m)}$ if $n \le m$.*

Theorem 1 can be proved by the following lemmas.

Lemma 2. *Let fc be the function defined as follows:*

$$fc(n) \stackrel{\text{def}}{=} \begin{cases} n(n-1)/2 & \text{if } n \le c, \\ \displaystyle\sum_{i=0}^{c-1} \{fc(\lfloor (n+i)/c \rfloor) + (c-i-1)\lfloor (n+i)/c \rfloor\} & \text{otherwise.} \end{cases}$$

For all $n \ge 1$, $Cc_{(n)}$ has $fc(n)$ edges.

Lemma 3. *Let gc be the function defined as follows:*

$$gc(n) \stackrel{\text{def}}{=} \begin{cases} n(n-1)/2 & \text{if } n \le c, \\ \max\{\sum_{i=0}^{c-1}\{gc(n_i) + (c-i-1)n_i\} \mid & \\ \quad n_0 \le n_1 \le \cdots \le n_{c-1} < n = \sum_{i=0}^{c-1} n_i\} & \text{otherwise.} \end{cases}$$

For any subgraph $G = (V, E)$ of $Cc_{(m)}$, $|E| \le gc(|V|)$ holds.

Lemma 4. *$fc = gc$ holds.*

Note that the division of an integer n into the same c values as equally as possible can be represented as

$$\lfloor n/c \rfloor, \lfloor (n+1)/c \rfloor, \lfloor (n+2)/c \rfloor, \ldots, \lfloor (n+c-1)/c \rfloor.$$

In fact, the sequence is $c - r$ q's followed by r $(q+1)$'s where $n = q \cdot c + r$ $(0 \le r \le c-1)$. Thus, while $gc(n)$ is evaluated by computing the maximum over all divisions of n, $fc(n)$ is evaluated for the equal-sized division of n. Therefore, obviously, we have $fc \le gc$. However, Lemma 4 claims $fc = gc$.

Lemma 2 shows the number of edges of $Cc_{(n)}$, and Lemma 3 shows the upper bound of the number of edges of the maximum subgraph. Hence, from Lemma 4, the number of edges of $Cc_{(n)}$ is equal to the number of edges of the maximum subgraph with n nodes. Therefore, these lemmas imply Theorem 1. See the appendix for the proofs of Lemmas 2, 3, and 4.

3 Widths and length of Cc_d

To get exact evaluations of the widths of Cc_d, we first prove the following lemma:

Lemma 5. *For any linear layout L and any gap i $(1 \le i \le c^d - 1)$, the cut of Cc_d under L at i is at least as large as that of Cc_d under the label order layout at i.*

Proof. For a gap i under L, divide the edges in Cc_d into $Cc_d^-(L,i)$, $Cc_d^+(L,i)$, and $Cc_d(L,i)$ as follows: $Cc_d^-(L,i)$ (resp. $Cc_d^+(L,i)$) is the set of edges connecting nodes whose positions are less than (resp. larger than or equal to) i, and $Cc_d(L,i)$ are the cut under L at a gap i. Obviously, we have

1. $Cc_d^-(L,i)$ and $Cc_d^+(L,i)$ are subgraphs of Cc_d with i nodes and with $c^d - i$ nodes, respectively.
2. Let I be the label order layout, i.e. for all i, $I(i) = i$ (Fig 3). Since the label order layout of Cc_d is bilateral symmetry, $Cc_d^-(I,i)$ and $Cc_d^+(I,i)$ correspond to the edges of $Cc_{(i)}$ and $Cc_{(c^d-i)}$, respectively.

Hence, from Theorem 1, $|Cc_d^-(L,i)| \le |Cc_d^-(I,i)|$ and $|Cc_d^+(L,i)| \le |Cc_d^+(I,i)|$ hold. Furthermore, obviously,

$$|Cc_d(L,i)| + |Cc_d^-(L,i)| + |Cc_d^+(L,i)| = |Cc_d(I,i)| + |Cc_d^-(I,i)| + |Cc_d^+(I,i)|.$$

Thus, $|Cc_d(L,i)| \ge |Cc_d(I,i)|$ holds. This completes the proof. □

From this lemma, when computing the parameters of Cc_d, we do not have to compute the minimum over all linear layouts but only those of the label order layout. In other words, we have

Lemma 6. *The bisection width, cut width, and total edge length of Cc_d are equal to those of the label order layout, respectively.*

It is easy to compute the parameters of the label order layout of Cc_d. For example, the bisection width of the label order layout (i.e. $|Cc_d(I, \lfloor c^d/2 \rfloor)|$) can be computed as follows: If c is even, since $Cc_d(I, \lfloor c^d/2 \rfloor)$ consists of edges along the dth dimension, $|Cc_d(I, \lfloor c^d/2 \rfloor)|$ is equal to $c^d/2 \times c/2 = c^{d+1}/4$. If c is odd, among all edges along each kth dimension ($1 \le k \le d$), $Cc_d(I, \lfloor c^d/2 \rfloor)$ contains $(c^2-1)c^{k-1}/4$ edges. By summing up, $Cc_d(I, \lfloor c^d/2 \rfloor)$ has $(c+1)(c^d-1)/4$ edges.

As a result, we have the following theorem:

Theorem 7. *The bisection width of Cc_d is $c^{d+1}/4$ (if c is even), and $(c+1)(c^d-1)/4$ (if c is odd).*

Similarly, we can compute the cut width and total edge length of the label order layout and get the following theorems:

Theorem 8. *The cut width of Cc_d is $c(c+2)(c^d-1)/\{4(c+1)\}$ (if c is even and d is even), $c^2\{(c+2)c^{d-1}-1\}/\{4(c+1)\}$ (if c is even and d is odd), and $(c+1)(c^d-1)/4$ (if c is odd).*

Theorem 9. *The total edge length of Cc_d is $(c+1)c^d(c^d-1)/6$.*

4 Widths and length of Ac_d

Since $Ac_{(n)}$ is not always a maximum subgraph, the method in the previous section cannot be applied to compute the widths of Ac_d. Hence, we use a method similar to embedding a directed clique [7]. In other words, Cc_d is embedded in Ac_d.

From Theorem 7, the bisection width of a c-node clique is $h(c)$, where $h(c)$ is $c^2/4$ (if c is even) and $(c^2-1)/4$ (if c is odd). Since each side of Cc_d can be considered as a c-node clique, we have

Lemma 10. *For any linear layout L and any gap i ($1 \le i \le c^d$), the cut of Cc_d under the label order layout at i is at most $h(c)$ times as large as the cut of Ac_d under L at i.*

Proof. Fix a linear layout L and compare L of Ac_d and L of Cc_d. It can be considered that each edge in Ac_d corresponds to at most $h(c)$ edges in Cc_d under L. Therefore, the cut of Cc_d under L at each gap is at most $h(c)$ times as large as the cut of Ac_d under L at the same gap. Thus, from Lemma 5, the cut of Cc_d under the label order layout at each position is at most $h(c)$ times as large as the cut of Ac_d under L at the same position. □

From this lemma, we have

Lemma 11. *The bisection width, cut width, and total edge length of Ac_d are at least as large as those of Cc_d divided by $h(c)$, respectively.*

Therefore, the lower bounds of Ac_d can be obtained from Theorems 7, 8, and 9.
On the other hand, from the definition, we have

Lemma 12. *The bisection width, cut width, and total edge length of Ac_d is at most as large as those of Ac_d under the label order layout, respectively.*

From these relation, the upper bounds of Ac_d can be obtained by computing those of the label order layout which can be computed similarly to those of Cc_d. Consequently, we have

Theorem 13. *The bisection width of Ac_d is c^{d-1} (if c is even), and $(c^d-1)/(c-1)$ (if c is odd).*

Theorem 14. *The cut width of Ac_d ($c \geq 3$) is at least $(c+2)(c^d-1)/\{c(c+1)\}$ (if c is even and d is even), at least $\{(c+2)c^{d-1}-1\}/(c+1)$ (if c is even and d is odd), at least $(c^d-1)/(c-1)$ (if c is odd), and at most $(c^d-1)/(c-1)$.*

If $c = 2$, the cut width of Ac_d is equal to that of Cc_d.

Theorem 15. *The total edge length of Ac_d is at least $2(c+1)c^{d-2}(c^d-1)/3$ (if c is even), at least $2c^d(c^d-1)/\{3(c-1)\}$ (if c is odd), and at most $c^{d-1}(c^d-1)$.*

Fortunately, the upper bound of the bisection width is equal to the lower bound. However, the upper bounds of the cut width and total edge length of Ac_d do not match the lower bounds of them. But the difference is not so large; The upper bound of the cut width is at most approximately $1+2/\{(c+2)(c-1)\}$ times as large as the lower bound and the upper bound of the total edge length is approximately 1.5 times as large as the lower bound.

5 Conclusions

We have presented the exact or nearly exact values of the bisection width, cut width, total edge length of generalized hypercubes. Lemma 5 implies that the label order layout of a d-dimensional c-ary clique is the optimal layout in the sense that the cut of the label order layout at each gap is smaller than or equal to that of any other layout at the same gap. Similarly to the Ac_d case, this result makes it easy to prove that the upper and lower bounds of the widths and the total length of Ac_d with wraparound edges (referred to as a d-dimensional c-ary torus [7]) are twice as large as those of Ac_d. The exact values of the cut width and total edge length of Ac_d remain to be solved.

Acknowledgment

The author would like to thank Imrich Vrťo for his valuable comments.

References

1. L. N. Bhuyan and D. P. Agrawal. Generalized hypercube and hyperbus structures for a computer network. *IEEE Transactions on Computers*, C-33(4), April 1984.
2. G. Brebner. Relating routing and two-dimensinal grids. In P. Bertolazii and F. Luccio, editors, *VLSI: Algorithms and Architectures*, pages 221–231. Elsevier Science Publishers B.V.(North-Holland), 1985.
3. R. A. DeMillo, S. C. Eisenstat, and R. J. Lipton. Preserving average proximity in arrays. *Communications of the ACM*, 21(3):228–231, March 1978.
4. M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified polynomial complete problems. *SIGACT*, pages 47–63, 1974.
5. L. H. Harper. Optimal assignments of numbers to vertices. *J. Soc. Indust. Appl. Math*, 12(1):131–135, March 1964.
6. F. T. Leighton. *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks.* MIT Press, 1983.
7. F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes.* Morgan Kaufmann, 1992.
8. F. Makedon and I. H. Subdorough. On minimizing width in linear layouts. *Discrete Applied Mathematics*, 23:243–265, 1989.
9. Y. Manabe, K. Hagihara, and N.Tokura. The minimum bisection widths of the cube-connected-cycles graph and cube graph. *Trans. IEICE(D) Japan*, J76-D(6):647–654, June 1984. in Japanese.
10. K. Nakano, W. Chen, T. Masuzawa, K. Hagihara, and N. Tokura. Cut width and bisection width of hypercube graph. *IEICE Transactions*, J73-A(4):856–862, April 1990. in Japanese.
11. L. Niepel and P. Tomasta. Elevation of a graph. *Czechoslovak Mathematical Journal*, 31(106):475–483, 1981.
12. A. L. Rosenberg. Preserving proximity in arrays. *SIAM J. Comput.*, 4(4):443–460, December 1975.
13. C. D. Thompson. Area-time complexity for VLSI. In *Proc. of 11th Symposium on Theory of Computing*, pages 81–88. ACM, 1979.
14. K. Wada and K. Kawaguchi. Optimal bounds of the crossing number and the bisection width for generalized hypercube graphs. In *Proc. of 16th Biennial Symposium on Communications*, pages 323–326, May 1992.
15. K. Wada, H. Suzuki, and K. Kawaguchi. The crossing number of hypercube graphs. In *Proc. of 43rd Convention of IPS Japan*, pages 1–95, 1991. in Japanese.

Appendix

In the appendix, we will prove Lemmas 2, 3, and 4. First, we will prove Lemma 2.

Proof. The proof is by induction on n. Obviously for all $n \leq c$, $Cc_{(n)}$ has $fc(n)$ edges. We assume that for all $k \leq n-1$, $Cc_{(k)}$ has $fc(k)$ edges, and will prove that $Cc_{(n)}$ has $fc(n)$ edges. For all i $(0 \leq i \leq c-1)$, let $Vc^i_{(n)}$ be the set of nodes such that the LSD's (Least Significant Digits) of the c-ary representations of them are $c-i-1$. In other words, $Vc^i_{(n)}$ contains the nodes $[\cdots \overset{LSD}{(c-i-1)}]$. Hence, $Vc^i_{(n)}$ consists of $\lfloor (n+i)/c \rfloor$ nodes. Let $Ec^{ij}_{(n)}$ $(i \leq j)$ be the edges connecting $Vc^i_{(n)}$ and $Vc^j_{(n)}$. Since for all i, $Gc^{ii}_{(n)} = (Vc^i_{(n)}, Ec^{ii}_{(n)})$ and $Cc_{(\lfloor (n+i)/c \rfloor)}$ are isomorphic, we have $|Ec^{ii}_{(n)}| = fc(\lfloor (n+i)/c \rfloor)$ from the inductive assumption. For all i and j $(i < j)$, no two edges in $Ec^{ij}_{(n)}$ share a node in $Vc^i_{(n)}$ and every node in $Vc^i_{(n)}$ is connected by an edge in $Ec^{ij}_{(n)}$. Thus $|Ec^{ij}_{(n)}| = |Vc^i_{(n)}| = \lfloor (n+i)/c \rfloor$. Therefore, we have

$$\begin{aligned} |Ec_{(n)}| &= \sum_{i=0}^{c-1} |Ec^{ii}_{(n)}| + \sum_{i<j} |Ec^{ij}_{(n)}| = \sum_{i=0}^{c-1} fc(\lfloor (n+i)/c \rfloor) + \sum_{i<j} \lfloor (n+i)/c \rfloor \\ &= \sum_{i=0}^{c-1} \{ fc(\lfloor (n+i)/c \rfloor) + (c-i-1)\lfloor (n+i)/c \rfloor \} = fc(n). \end{aligned}$$

□

Secondly, we will show the proof of Lemma 3.

Proof. The proof is by induction on the number of nodes in V. Obviously, for any subgraph $G = (V, E)$, if $|V| \leq c$ then $|E| \leq gc(|V|)$. We assume that $|E| \leq gc(|V|)$ if $|V| \leq n-1$, and will show that $|E| \leq gc(|V|)$ if $|V| = n$. We select any digit s and divide V into $V^0, V^1, \ldots, V^{c-1}$ as follows: V^i consists of the nodes such that the sth digit of the c-ary representation of them is i. In other words, V^i contains the nodes $[\cdots \overset{s\text{th digit}}{i} \cdots]$. Since we can select s such that there are at least two $V's$ which are not empty, we can assume, for all i, $|V^i| < |V|$. Furthermore, by renumbering the indices of $V's$, we can assume that $|V^0| \leq |V^1| \leq \cdots \leq |V^{c-1}| < |V|$ without loss of generality. Let E^{ij} $(i \leq j)$ be the edges in E connecting V^i and V^j. Since $|V^0| \leq |V^1| \leq \cdots \leq |V^{c-1}| < n$, we have, for all i, $E^{ii} \leq gc(|V^i|)$ from the inductive assumption. Since no two edges in E^{ij} $(i < j)$ share a node in V^i, $|E^{ij}|$ is at most as large as $|V^i|$. Therefore, we have

$$\begin{aligned} |E| &= \sum_{i=0}^{c-1} |E^{ii}| + \sum_{i<j} |E^{ij}| \leq \sum_{i=0}^{c-1} gc(|V^i|) + \sum_{i<j} |V^i| \\ &\leq \sum_{i=0}^{c-1} \{ gc(|V^i|) + (n-i-1)|V^i| \} \leq gc(|V|). \end{aligned}$$

□

We have to prove several lemmas as preparation for the proof of Lemma 4. From now on, for given $n_0, n_1, \ldots, n_{c-1}$, let $n = n_0 + n_1 + \cdots + n_{c-1}$ and $n_i = q_i c + r_i$ $(0 \leq r_i \leq c-1)$. For convenience, let $n_{-1} = -\infty$ and $n_c = +\infty$. Under this notation, the following lemma holds obviously:

Lemma 16. *For all* $n_0, n_1, \ldots, n_{c-1}$,

$$n = \sum_{j=0}^{c-1} \lfloor \frac{n+j}{c} \rfloor = \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} \lfloor \frac{n_i+j}{c} \rfloor .$$

Let us consider that for given $n_0, n_1, \ldots, n_{c-1}$ $(n_0 \le n_1 \le \cdots \le n_{c-1})$, the set $\{(i,j) | 0 \le i,j \le c-1\}$ is sorted by $\lfloor (n_i+j)/c \rfloor$, and let the $(ic+j)$th $(0 \le i,j \le c-1)$ smallest element[1] and its value be $(a(i,j), b(i,j))$ and $\alpha(i,j)$, respectively. In other words, for all i and j, let $\alpha(i,j) = \lfloor (n_{a(i,j)} + b(i,j))/c \rfloor$, and for all i, j, i', j' such that $ic+j \le i'c+j'$, we find that $\alpha(i,j) \le \alpha(i',j')$ holds. Figure 5 illustrates an example of the values of $\lfloor (n_i+j)/c \rfloor$ and $\alpha(i,j)$.

Consider the following procedure that determines two mappings $A, B : \{0, 1, \cdots, c-1\} \times \{0, 1, \cdots, c-1\} \rightarrow \{0, 1, \cdots, c-1\}$:

Step 1 Let $S_j := \lfloor (n+j)/c \rfloor$ for all j $(0 \le j \le c-1)$, and $i := 0$.
Step 2 Sort $S_0, S_1, \ldots, S_{c-1}$ by their values and let $p : \{0, \ldots, c-1\} \rightarrow \{0, \ldots, c-1\}$ be the one-to-one mapping so that for each j, $S_{p(j)}$ is the jth smallest element.
Step 3 For all j, determine A and B so that $A(i, p(j)) = a(i,j)$ and $B(i, p(j)) = b(i,j)$.
Step 4 For all j, let $S_{p(j)} := S_{p(j)} - \alpha(i,j)$.
Step 5 Let $i := i+1$ and if $i < c-1$ then go to Step 2.

				$\lfloor (n_i+j)/c \rfloor$								$\alpha(i,j)$								$\beta(i,j)$							
$i \backslash j$	n_i	q_i	r_i	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
0	21	2	5	2	2	2	3	3	3	3	3	2	2	2	3	3	3	3	3	2	2	2	3	3	3	3	3
1	34	4	2	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	36	4	4	4	4	4	4	5	5	5	5	4	4	5	5	5	5	5	5	5	5	5	4	4	5	5	5
3	57	7	1	7	7	7	7	7	7	7	8	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	60	7	4	7	7	7	7	8	8	8	8	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	60	7	4	7	7	7	7	8	8	8	8	7	7	8	8	8	8	8	8	7	8	8	8	8	7	8	8
6	61	7	5	7	7	7	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
7	65	8	1	8	8	8	8	8	8	8	9	8	8	8	8	8	8	8	9	9	8	8	8	8	8	8	8
																			SUM	49	49	49	49	49	49	50	50

Fig. 5. An example of the values of $\lfloor (n_i+j)/c \rfloor$, $\alpha(i,j)$, and $\beta(i,j)$

Since A and B are determined one by one from the smallest to the largest element of $\{(i,j) | 0 \le i,j \le c-1\}$, A and B have the following two properties after completion of the procedure.

Property 17. $C(i,j) = \langle A(i,j), B(i,j) \rangle$ *is a one-to-one mapping.*

[1] Let the 0th smallest element be the smallest element.

Property 18. *Let* $\beta(i,j) = \lfloor (n_{A(i,j)} + B(i,j))/c \rfloor$. *For all* j, $\beta(0,j) \le \beta(1,j) \le \cdots \le \beta(c-1,j)$.

These properties can be clearly seen in Fig. 5.

For all i, $q_i \le \alpha(i,0) \le \alpha(i,1) \le \cdots \le \alpha(i,c-1) \le q_i + 1$ holds. Hence, after each iteration, for all j and j', $|S_j - S_{j'}| \le 1$ holds. In particular, from Lemma 16 and Property 17, for all j, $S_j = 0$ after completion of the procedure. Therefore, β has the following property:

Property 19. *For all* j,

$$\sum_{i=0}^{c-1} \beta(i,j) = \lfloor \frac{n+j}{c} \rfloor.$$

Let $\sum_{i=s}^{t} r_i = q_{s,t}c + r_{s,t}$ $(0 \le r_{s,t} \le c-1)$. For all s,t such that $q_{s-1} < q_s = q_{s+1} = \cdots = q_t < q_{t+1}$, let us imagine the three submatrices which can be obtained by picking up from the s-th to the t-th row of the matrices in Fig. 5. For example, choose $s = 3$ and $t = 6$ in Fig. 5. The submatrices of $\alpha(i,j)$, $\beta(i,j)$, and $\lfloor (n_i + j)/c \rfloor$ have the following property:

Property 20.
- *They have the same number of* q_s *'s and the same number of* $q_s + 1$ *'s.*
- *In the ith row of the submatrix of* $\lfloor (n_i+j)/c \rfloor$, *there are* $(c - r_i)$ q_s *'s followed by* r_i $q_s + 1$ *'s.*
- *In the* $(t - q_{s,t})$*th row of the submatrix of* $\alpha(i,j)$, *there are* $(c - r_{s,t})$ q_s *'s followed by* $r_{s,t}$ $q_{s,t} + 1$ *'s, and the rows above and below are filled with* q_s *'s and* $q_s + 1$ *'s, respectively.*
- *In the* $(t - q_{s,t})$*th row of the submatrix of* $\beta(i,j)$, *there are* $(c - r_{s,t})$ q_s *'s and* $r_{s,t}$ $q_{s,t} + 1$ *'s, and the rows above and below filled with* q_s *'s and* $q_s + 1$ *'s, respectively.*

From Property 20, we have

Lemma 21. *For all* s *and* t *such that* $q_{s-1} < q_s = q_{s+1} = \cdots = q_t < q_{t+1}$,

$$\sum_{i=s}^{t} \sum_{j=0}^{c-1} (i+j)\beta(i,j) \le \sum_{i=s}^{t} \sum_{j=0}^{c-1} (i+j)\alpha(i,j) \le \sum_{i=s}^{t} \sum_{j=0}^{c-1} (i+j) \lfloor \frac{n_i + j}{c} \rfloor.$$

From Lemma 21, we have the following corollary:

Corollary 22.

$$\sum_{i=0}^{c-1} \sum_{j=0}^{c-1} (i+j)\beta(i,j) \le \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} (i+j)\beta(i,j) \le \sum_{i=0}^{c-1} \sum_{j=0}^{c-1} (i+j) \lfloor \frac{n_i + j}{c} \rfloor.$$

Now, we will prove Lemma 4.

Proof. Since $fc \leq gc$ from the definition, it suffices for the lemma to prove $fc \geq gc$. We prove $fc \geq gc$ by induction. Obviously, $fc(n) = gc(n)$ if $n \leq c$. We assume that for all $i\,(< n)$, $fc(i) \geq gc(i)$ holds, and will prove $fc(n) \geq gc(n)$. For all $n_0, n_1, \ldots, n_{c-1}$ such that $n_0 \leq n_1 \leq \ldots \leq n_{c-1} < n$ and $n > c$, the following relation holds:

$$\begin{aligned}
&\sum_{i=0}^{c-1}\{gc(n_i) + (c-i-1)n_i\} \\
&\leq \sum_{i=0}^{c-1} fc(n_i) + \sum_{i=0}^{c-1}(c-i-1)n_i \qquad \text{(from the inductive assumption)} \\
&= \sum_{i=0}^{c-1}\sum_{j=0}^{c-1}\{fc(\lfloor\frac{n_i+j}{c}\rfloor) + (c-j-1)\lfloor\frac{n_i+j}{c}\rfloor\} + \sum_{i=0}^{c-1}(c-i-1)n_i \\
&= \sum_{i=0}^{c-1}\sum_{j=0}^{c-1} fc(\lfloor\frac{n_i+j}{c}\rfloor) + (2c-2)n - \sum_{i=0}^{c-1}\sum_{j=0}^{c-1}(i+j)\lfloor\frac{n_i+j}{c}\rfloor \\
&\qquad \text{(from Lemma 16)}
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
fc(n) &= \sum_{j=0}^{c-1}\{fc(\lfloor\frac{n+j}{c}\rfloor) + (c-j-1)\lfloor\frac{n+j}{c}\rfloor\} \\
&\geq \sum_{j=0}^{c-1} gc(\frac{n+j}{c}) + \sum_{j=0}^{c-1}(c-j-1)\lfloor\frac{n+j}{c}\rfloor \\
&\qquad \text{(from the inductive assumption)} \\
&\geq \sum_{i=0}^{c-1}\sum_{j=0}^{c-1}\{gc(\beta(i,j)) + (c-i-1)\beta(i,j)\} + \sum_{i=0}^{c-1}\sum_{j=0}^{c-1}(c-j-1)\beta(i,j) \\
&\qquad \text{(from Properties 18 and 19)} \\
&= \sum_{i=0}^{c-1}\sum_{j=0}^{c-1} gc(\lfloor\frac{n_i+j}{c}\rfloor) + (2c-2)n - \sum_{i=0}^{c-1}\sum_{j=0}^{c-1}(i+j)\beta(i,j). \\
&\qquad \text{(from Lemma 16 and Property 17)}
\end{aligned}$$

Thus, from $fc \leq gc$ and Corollary 22, we have:

$$\sum_{i=0}^{c-1}\{gc(n_i) + (c-i-1)n_i\} \leq fc(n).$$

Therefore, $gc(n) \leq fc(n)$ holds. □