Graph Theory: Applications and Algorithms

CIS008-2 Logic and Foundations of Mathematics

David Goodwin

david.goodwin@perisic.com

University of Bedfordshire

11:00, Tuesday 21st February 2012

Outline

Outline

1. *n*-Cube
2. Gray Codes
3. Shortest-Path Algorithm
4. Mesh Model

University of Bedfordshire

THE *n*-CUBE (HYPERCUBE)

- **Serial Computers** execute one instruction at a time.
- Standard definition of an algorithm assumes one instruction is exectuted at a time, and is called a **serial algorithm**.
- **Parallel computers** execute several instructions at a time, and their associated algorithms are **parallel algorithms**.
- Graphs are convenient models for parallel computation, and one model for this is known as the *n*-**Cube** or **hypercube**.

The n-Cube (Hypercube)

- n-Cube has 2^n processors, represented by vertices, labelled $0, 1, \ldots, 2^n - 1$. Each processor has its own memory.

The *n*-Cube (Hypercube)

- *n*-Cube has 2^n processors, represented by vertices, labelled $0, 1, \ldots, 2^n - 1$. Each processor has its own memory.
- An edge is drawn between two vertices if the binary representation of the labels differs in exactly one bit.

The *n*-Cube (Hypercube)

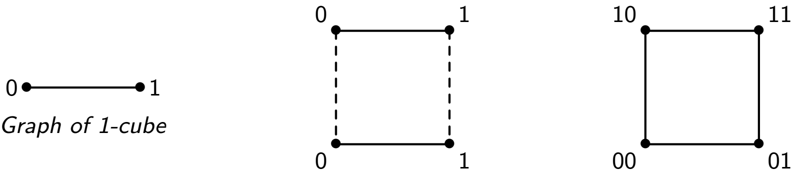
- *n*-Cube has 2^n processors, represented by vertices, labelled $0, 1, \ldots, 2^n - 1$. Each processor has its own memory.
- An edge is drawn between two vertices if the binary representation of the labels differs in exactly one bit.
- During one time unit, all processors may execute an instructionsimultaneously, then communicate with an adjacent processor.

The n-Cube (Hypercube)

- n-Cube has 2^n processors, represented by vertices, labelled $0, 1, \ldots, 2^n - 1$. Each processor has its own memory.
- An edge is drawn between two vertices if the binary representation of the labels differs in exactly one bit.
- During one time unit, all processors may execute an instructionsimultaneously, then communicate with an adjacent processor.
- To communicate with a non-adjacent processor, a message is sent that includes the route to and final destination of the recipient. It may take several time units to communicate with a non-adjacent processor.

The *n*-Cube (Hypercube)

- The 1-cube has two vertices (processors), labelled 0 and 1, and one edge.
- We draw two $(n-1)$-Cubes, whose vertices are labelled in binary.
- We place an edge between vertices with identical labels.
- We then place an additional character at the beginning of each of the two graphs, 0 at the front of one, and 1 at the font of the other's labels.
- Here we obtain an n-Cube from two $(n-1)$-Cubes.



Graph of 1-cube *Connecting two 1-cubes* *Graph of 2-cube*

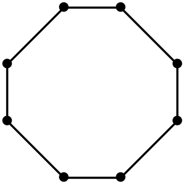
Outline

1. n-Cube
2. Gray Codes
3. Shortest-Path Algorithm
4. Mesh Model

University of Bedfordshire

Gray codes and Hamiltonian cycles in the *n*-Cube

- Consider a **ring model** for parallel computation, represented as a graph is a simple cycle.
- Each processor can communicate directly with two other processors.
- The *n*-Cube model has a greater degree of connectivity between vertices.
- If the *n*-Cube contains a Hamiltonian cycle, we must have $n \geq 2$, since the 1-Cube has no cycles.



Ring model

University of Bedfordshire

Gray Codes

- The n-Cube has a Hamiltonian cycles if and only if $n \geq 2$ and there is a sequence,

$$s_1, s_2, \ldots, s_{2^n}$$

  where each s_i is a string of n bits satisfying:
  - Every n-bit string appears somewhere in the sequence.
  - s_i and s_{i+1} differ in exactly one bit.
  - s_{2^n} and s_1 differ in exactly one bit.
- This type of sequence is called a **Gray code**, and corresponds to a Hamiltonain cycle.
- Gray codes have been studied extensively, an example being converting analog to digital signal.
- The Gray code for $n = 1$ is $0, 1$ and corresponds to the path $(0, 1, 0)$, which is not a cycle since the edge $(0, 1)$ is repeated.

Constructing Gray codes

Let G_1 denote the sequence $0, 1$, we define G_n by G_{n-1} by the following rules:

1. Let G^R_{n-1} denote the sequence G_{n-1} in reverse.
2. Let G'_{n-1} denote the sequence obtained by prefixing each member of G_{n-1} with 0.
3. Let G''_{n-1} denote the sequence obtained by prefixing each member of G_{n-1} with 1.
4. Let G_n be the sequence consisting of G'_{n-1} followed by G''_{n-1}

Then G_n is a Gray code for every positive integer n.

Theorem

The n-Cube has a Hamiltonian cycle for every positive integer $n \geq 2$.

University of Bedfordshire

Outline

Dijkstra's Shortest Path Algorithm

- The aim is to find the shortest path (having the minimum length) between any two vertices, a and z, of a connected, weighted graph, G.
- Dijkstra's Algorithm involves assigning lables to vertices. Let $L(u)$ denote the label of vertex v.
- At any point, some vertices have temporary labels and the rest have permanent labels.
- Let t denote the set of vertices with temporary labels.
- Initially, all vertices have temporary labels.
- Each iteration of the algorithm changes the status of one label to permanent.
- The algorithm terminates when z recieves a permanent label, then $L(z)$ gives the length of the shortest path from a to z.

Input: A connected, weighted graoh in which all weights are positive; vertices a and z.
Output: $L(z)$, the length of a shortest path from a to z.

$dijkstra(w, a, z, L)\{$

$L(a) = 0$

for all vertices $x \neq a$

    $L(x) = \infty$

$T =$ set of all vertices

// T is the set of all vertices whose shortest distance has not been found

while$(z \in T)\{$

    chose $v \in T$ with minimum $L(v)$

    $T = T - \{v\}$

    for each $x \in T$ adjacent to v

        $L(x) = min\{L(x), L(v) + w(v, x)\}$

    $\}$

$\}$

Path length and the Adjacency matrix

Theorem

If A is the adjacency matrix of a simple graph, the ij^{th} entry of A^n is equal to the number of paths of length n from vertex i to vertex j, $n = 1, 2, \ldots$.

Outline

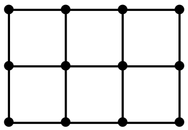
1. *n*-Cube
2. Gray Codes
3. Shortest-Path Algorithm
4. Mesh Model

ISOMORPHISMS OF GRAPHS

Graphs G_1 and G_2 are **isomorphic** if there is a one-to-one, onto function f from the vertices of G_1 to the vertices of G_2 and a one-to-one, onto function g from the edges of G_1 to the edges of G_2. So the edge e is incident on v and w in G_1 if and only if the edge $g(e)$ is incident on $f(v)$ and $f(w)$ in G_2. The pair of functions g and f is called an **isomorphism** of G_1 onto G_2.

The Mesh Model for Parallel Computation

- 2-D mesh model described as a graph consisting a rectangular array of vertices connected by edges to their nearest neighbour.
- Let M be a mesh p vertices by q vertices, where $p \leq 2^i$ and $q \leq 2^j$.
- As co-ordinates for the mesh, we use the Gray codes.
- The co-ordinates in the horizonatal direction are the first p member of an i-bit Gray code.
- The co-ordinates in the vertical direction are the first q member of an j-bit Gray code.



Mesh Model

EXERCISE Show that the mesh M, with p by q vertices ($p \leq 2^i$ and $q \leq 2^j$) has an isomorphism to a subgraph contained in the $(i + j)$-cube.