P2P and Grid Computing
- Exercise 5 -
- C110 -

Michael Scholz (Matr. 1576630)
Ulf Gebhardt (Matr. 1574373)

H 5.1:

a)

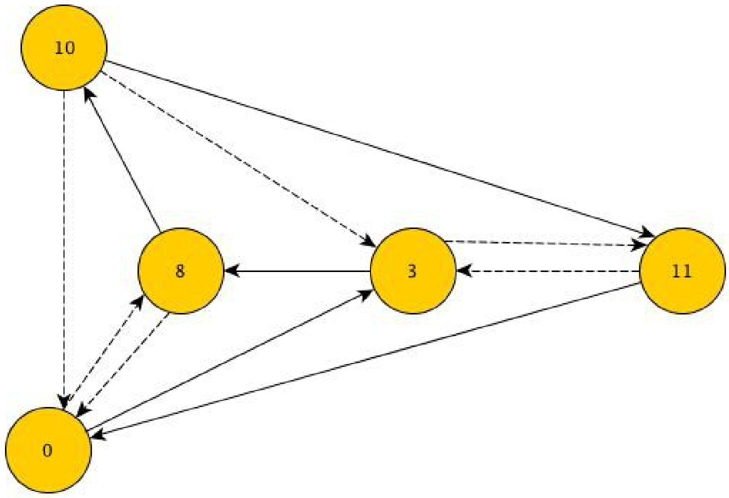

Tabelle für **11**:

Start	Int.	Succ
12	[12,13)	0
13	[13,15)	0
15	[15,3)	0
3	[3,11)	3

Tabelle für **0**:

Start	Int.	Succ
1	[1,2)	3
2	[2,4)	3
4	[4,8)	8
8	[8,0)	8

Tabelle für **3**:

Start	Int.	Succ
4	[4,5)	8
5	[5,7)	8
7	[7,11)	8
11	[11,3)	11

Tabelle für **8**:

Start	Int.	Succ
9	[9,10)	10
10	[10,12)	10
12	[12,0)	0
0	[0,8)	0

Tabelle für **10**:

Start	Int.	Succ
11	[11,12)	11
12	[12,14)	0
14	[14,2)	0
2	[2,10)	3

b)

op	**0**	**3**	**8**	**10**	**11**
0	0	1	1	2	3
3	2	0	1	2	1
8	1	2	0	1	2
10	1	1	2	0	1
11	1	1	2	3	0

Charakteristische Pfadlänge:

$$\frac{1}{20} * 31 = 1.55$$

Ein Graph G heißt stark zusammenhängend, wenn es zwischen jedem beliebigen Knotenpaar u und v ein Pfad gibt. Dies ist bei Chord-Netzwerken der Fall, da diese immer eine Ringstruktur besitzen.

c)

op	**0**	**3**	**8**	**10**	**11**
0	0	1	1	2	1
3	1	0	1	2	1
8	1	1	0	1	2
10	1	1	1	0	1
11	1	1	2	1	0

durchschnittliche Routinglänge:

$$\frac{1}{20} * 24 = 1.2$$

Optimales Routing würde bedeuten, dass eine Nachricht zwischen zwei beliebigen Knoten über eine Kante geroutet werden würde. Dies ist bei Chord nicht der Fall. Hier werden Nachrichten über mehrere Knoten hinweg geroutet. Für ein optimales Routing müssten also alle Knoten miteinander verbunden sein. Dies würde jedoch auch zu einem großen Speicherverbrauch führen, da jeder Knoten die Informationen über jeden übrigen Teilnehmer vorhalten müsste.

H 5.2:

a)

Durchschnittliche Fall: O(log n)
Schlechtester Fall: O($\log^2$ n)
Im durchschnittlichen Fall finden wir den gesuchten Eintrag in O(log n). Dies kann man mit der binären Suche vergleichen. Die Datensätze werden in jedem Schritt ungefähr halbiert. Im schlechtesten Fall müssen alle Einträge in den Finger-Tabellen verfolgt werden.

b)

Knoten u muss durchschnittlich $\frac{k}{2}$ Anfragen an seine Nachbarn schicken.
Im schlechtesten Fall muss er k Anfragen an seine Nachbarn schicken.
Die Komplexitätsklasse ist hier O(n).

Wenn der Knoten jedes mal einen zufälligen Nachbarn wählt, so muss er durchschnittlich immernoch $\frac{k}{2}$ Anfragen an seine Nachbarn schicken. Dies liegt an der geometrischen Zufallsverteilung. Im schlechtesten Fall jedoch wählt Knoten u nie den richtigen Knoten. Somit wird keine Terminierung erreicht und der Knoten u stellt unendlich viele Anfragen an seine Nachbarn. Somit kann keine obere Schranke für die Komplexität angegeben werden. Es kann nur eine untere Schranke n/2 angegeben werden.

H 5.3:

a)

i: Maximale Distanz: Die Metrik beschreibt eine XOR-Verknüpfung der beiden Identifizierer. Diese bestehen aus n-bit. Somit ist der maximale Abstand $2^n - 1$.

ii: Maximale Distanz: $\frac{2^n}{2}$

b)

i: Zu einer vorgegebenen ID gibt es hier nur eine ID, die einen Abstand von 1 hat. Diese unterscheidet sich dann nur im untersten Bit von der vorgegebenen ID.

ii: Zu einer vorgegebenen ID gibt es hier zwei IDs, welche einen Abstand von 1 haben. Dies lässt sich intuitiv am Abstand zweier Zahlen auf dem Zahlenstrahl erklären: Die Betragsformel definiert den Abstand in beide „Richtungen“ auf dem Zahlenstrahl.

c)

Die erste Metrik lässt sich leicht in der Binärdarstellung ausrechnen. Für die zweite Metrik empfiehlt sich zuerst die Umformung in die Dezimaldarstellung.

i: $d_1(u, v) = 11\,111\,110_2 = 253_{10}$
$d_2(u, v) = 2_{10}$

i: $d_1(u, v) = 10\,000\,100_2 = 132_{10}$
$d_2(u, v) = 124_{10}$

d)

Nein. Es kann höchstens Gleichheit erreich werden. Dies ware zum Beispiel bei u = 00 und v = 01 der Fall.