Uebungsstunde vom 21.12.01, Info I

21. Dezember 2001

1 Nachbesprechung

1.1 Abbruchkriterium für String2LONGINT Konverter

Welches ist die richtige Abfrage, um einen overflow zu verhindern, d.h. alle Zahlen bis und mit 2147483647 zu akzeptieren, aber einen Trap zu vermeiden? Was gehört an die Stelle der Punkte?

```
MODULE Decoder;

        PROCEDURE ToInt(string:ARRAY OF CHAR):LONGINT;
                VAR result,i:LONGINT;overflow,letter:BOOLEAN;
                BEGIN
                        result:=0; i:=0;
                        letter:=FALSE; overflow:=FALSE;
                        WHILE (~overflow & ~letter & string[i]#0X) DO
                                IF ( ........ ) THEN overflow:=TRUE; END;
                                IF (string[i]<"0" OR string[i]>"9") THEN
                                        letter:=TRUE;
                                END;
                                result:=result*10 + (ORD(string[i])-ORD("0"));
                        END;
                        IF (overflow OR letter) THEN result:=-1; END;
                        RETURN result;
                END ToInt;

END Decoder.
```

1. IF (result>=MAX(LONGINT)) THEN ...
2. IF (i>=10) THEN ...
3. IF (result > (MAX(LONGINT)-(ORD(string[i])-ORD("0")) DIV 10) THEN ...
4. IF (result*10 + (ORD(string[i])-ORD("0"))>MAX(LONGINT))THEN ...
5. IF (result > 214748364 & (ORD(string[i])-ORD("0"))>7) THEN ...

1.2 Mergesort - Quicksort

```
PROCEDURE run*();
        VAR array:ARRAY len OF INTEGER;
        BEGIN
                array einlesen;
                merge(0,len-1,array);
                array ausgeben;
END run;

PROC merge(L,R:INTEGER; VAR array:ARRAY OF INTEGER);
        VAR k,i,j,m: INTEGER; temp: ARRAY len OF INTEGER;
        BEGIN
                m:=(R+L)DIV 2; i:=0; j:=m+1; k:=0;
```

1.3 Hashing

Hashformel: $((c_i * b * (i+1)) + ... + (c_1 * b * 2) + (c_0 * b * 1))$ MOD p

Welcher Parameter wird bei einer Kollision vergrössert?

2 Vorbesprechung

2.1 LeaderElection

Grundproblem: Alles identische Computer, wer übernimmt Koordination?

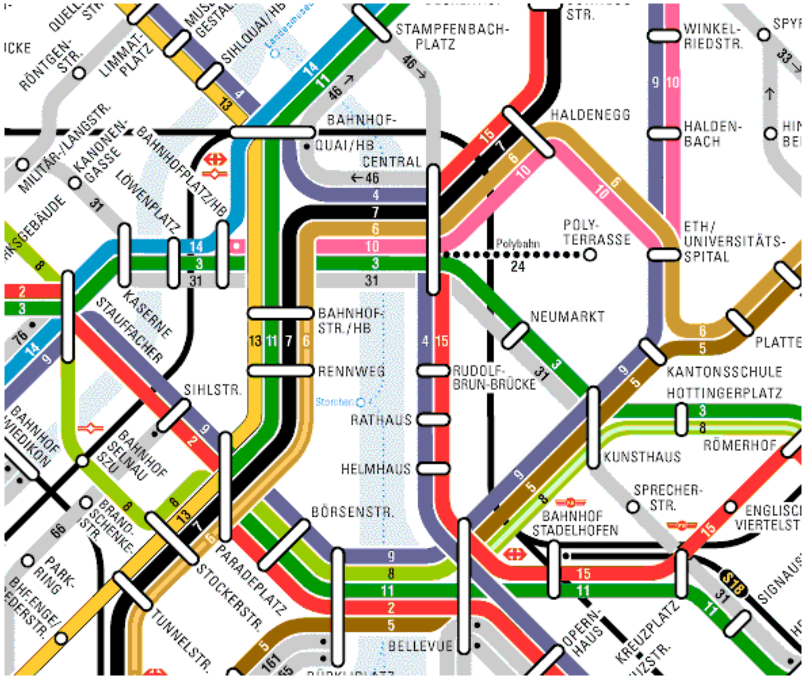
Start: jeder schickt seine Nummer an den Nachbarn im Uhrzeigersinn.
Was passiert mit der erhaltenen Nachricht:

- Erhaltene Nr $<$ eigene Nr:
- Erhaltene Nr $>$ eigene Nr:
- Erhaltene Nr $=$ eigene Nr:

Anordnung im Kreis in Computer repräsentieren: Zufällig in Array verteilen (= Permutation erzeugen)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

2.2 Elektronischer Fahrplan



Reise von Haldenegg zur Stockerstrasse ⇒Wie lange dauert die Fahrt auf dem optimalen Weg?

Abfahrtszeiten ab Haldenegg
Richtung Wollishofen

Gültig ab 10. Juni 2001

Ungefähre Reisezeiten in Minuten:

2 Central	10 Stockerstrasse	14 Brunaustrasse	19 Morgental
5 Bahnhofstrasse/Hauptbahnhof	11 Tunnelstrasse	15 Billoweg	20 Butzenstrasse
6 Rennweg	12 Bahnhof Enge	17 Bahnhof Wollishofen	21 Wollishofen
8 Paradeplatz	13 Museum Rietberg	18 Post Wollishofen	

Liste erstellen mit allen möglichen Verbindungen ab Haldenegg:

Start	Dest	Reisezeit	Linie
Haldenegg	ETH/Unispital	2	10
Haldenegg	ETH/Unispital	2	6
Haldenegg	Sonneggstr.	3	7
Haldenegg	Sonneggstr.	3	15
Haldenegg	Central	3	6
Haldenegg	Central	3	10
Haldenegg	Central	2	7
Haldenegg	Central	2	15

Rekursive Prozedur SwarmOut: Bei Ankunft an neuem Knoten:

- Ist das Ziel erreicht?
- Ist die Zeit kleiner als die bisher beste erreichte? ⇒Bestzeit eintragen, weiter SwarmOut
- Ist die Zeit grösser als die bisher beste erreichte? ⇒War wohl nichts, Sackgasse

Ablauf: Daten einlesen und strukturieren

- Liste generieren
- Knoten initialisieren
- Start/Ziel festlegen, Startschuss mit SwarmOut(Start, Ziel, 0)
- rekursives SwarmOut

Terminiert das Programm so?

Aenderung zur Aufgabenstellung: Umsteigezeit vernachlässigen sonst könnte ein Problem auftauchen:

und zum Schluss noch:

Schöne Weihnachten und einen guten Start ins neue Jahr!