CS 2110: Computer Programming Lab (Odd 2012)
Lab 11

Generating an Element in a Group

Instructor - John Augustine

November 5, 2012

Notes

- The 3-part series on graph algorithms has ended prematurely! This is in the interest of giving you a lab exercise on groups.
- Please follow the input/output format.
- This assignment is open ended (to some extent).
- Your solution must run in reasonable time.

Problem Definition

Consider the group $G = (\mathbb{Z}_{\geq 0}, +)$ where $+$ is the usual integer addition. The set $\{0, 1\}$ is a generating set for G, but you can also consider other generators. Note that 0 must be present in every generating set for G. Therefore, we do not need to explicitly state 0 — we can implicitly understand 0 to be in every generating set. Given a generating set $S \subseteq \mathbb{Z}_{\geq 0}$ and an $i \in \mathbb{Z}_{\geq 0}$, we can ask: how efficiently can we generate i using the elements in S. For example, let $S = \{1, 5\}$[1] and $i = 11$. You can generate 11 by the sequence $1 \to 2 \to 3 \to 4 \to \cdots \to 10$. More efficiently, you can generate 11 by $1 \to 2 \to 4 \to 8 \to 10 \to 11$. Optimally (i.e., most efficiently) you can generate 11 by $5 \to 10 \to 11$ or $1 \to 6 \to 11$. In general, you can define the efficiency of your generation by the number of times you use the $+$ operator. Thus, the problem is:

> Given a set $S \subseteq \mathbb{Z}_{\geq 0}$ and an element $i \in \mathbb{Z}_{\geq 0}$, produce a sequence of elements generated from S such that you reach i most efficiently (i.e., fewest number of times you use the $+$ operator).

(In this lab exercise, bulk of your grade is for getting your program to work when $S = \{1\}$. Therefore, first focus on that simple case. This will also help you get an intuition for how to solve the more general case when S can be an arbitrary subset of $\mathbb{Z}_{\geq 0}$.)

0.1 Command Line Arguments

Your command line argument for this assignment is a single input file name.

[1] Recall that 0 is implicitly in S

0.2 The Input File Format

Each input file should solve one instance of the problem. The format is as follows. First you must have an integer n which represents the number of elements of the generating set S. Then, you provide the elements in S one after another. Finally, you provide your value of i.

You should allow your input to be in free format. Confer Lab 10 for what I mean by free format.

You may assume that i and the largest element in S are never more than 100 and the cardinality of S is at most 10.

0.3 Output

The output should contain the following:

1. A sequence that leads to i.
2. The number of times you needed to use the + operator to reach the value i.

As an example, consider the input:

```
3 1 7 13 42
```

Then, the output format is:

```
13+7=20, 20+1 = 21, 21+21 = 42
Number of times we used the + operator = 3
```

The output format is not strict, but it should be very easy for the human reader to get the above two pieces of information.

0.4 Grading

There will be NO viva for this exercise. You must upload your source file by Friday (Nov 9) night (11:55PM). Getting your program to work for $S = \{1\}$ constitutes 80% of your marks. The rest 20% will be for getting your program to work for any S.

Since there is no viva component, we will check for plagiarism very carefully. You are allowed to discuss ideas with each other, but you must write your own program. And, more importantly, you must not share your program with others.

Uploading into MOODLE

Your code should be written as a single .c file. You must first compress the file using `gzip -c filename.c > filename.c.gz` and then the compressed .gz file must be uploaded into moodle. A link will be set up for this purpose in moodle.

Your TA for this lab

CS08B031, CS10B052, CS11B001 — CS11B009	Tejas Kulkarni
CS11B011 — CS11B021	Paresh Nakhe
CS11B022 — CS11B032	Shrikant Polawar
CS11B033 — CS11B042	Sai Sreenivas
CS11B043 — CS11B053	Nishaanth
CS11B054 — CS11B063	Saurav Kant Jha