Homework #1

Minimize resources
under latency constraint (MR-LC)

Date: 2023/10/20

Due: 2023/11/14 14:55

Homework #1

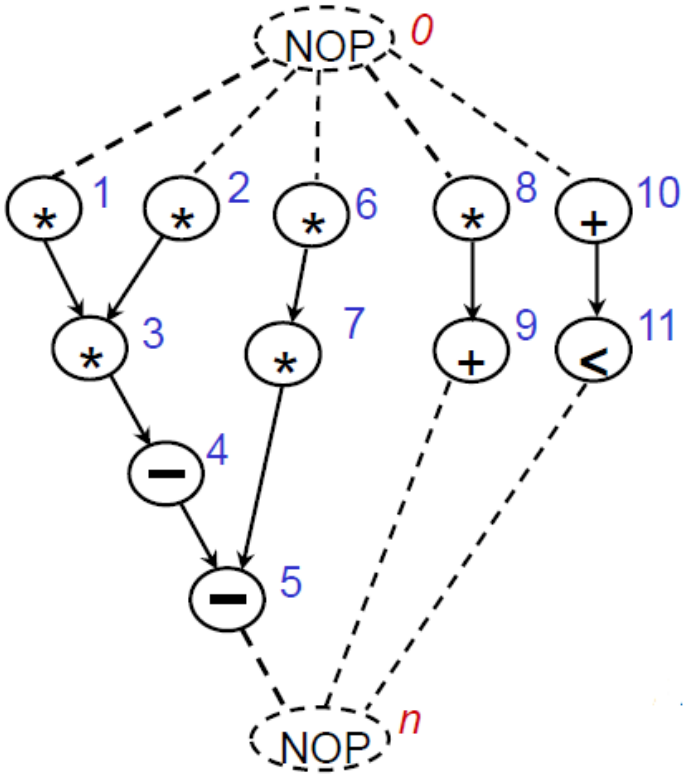
- **Due: 2023/11/14(WED.) 14:55**

- Language
  - Please use C++ language to implement your program.

- Program
  - Use List Scheduling algorithm to solve a latency-constrained resources minimization (MR-LC) problem.
  - Please refer to lecture note ‘High Level Synthesis (I)’ pp.61-66 for more information.
  - Assume that all operations have unit delay.

Homework #1

- Problem statement
  - You are given a sequencing graph consisting of v nodes including two **NOP** nodes which are the source and sink.
  - The nodes are numbered from 0 to $n = v - 1$, the source is the node 0 and the sink is node n.
  - Each node represents an operation.
  - All operations have unit delay.
  - What's the minimum usage of resources of this graph under a specified latency constraint?



Homework #1

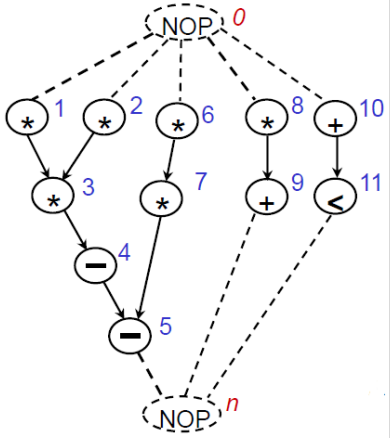
- Input file
  - Input files are named as BenchX.dfg.
  - The number of nodes v and latency constraint L ($3 \leq v \leq 50, 1 \leq L \leq 10$) are provided in the first and second line, respectively.
  - v lines follow. Each line contains a list of integers.The first is the index of the node, followed by the resource type it takes (0 for NOP, 1 for MULT, and 2 for ALU), and then the number of its successors. The rest represents the successor(s) of the node.
- Output file
  - Output your report to a file named as BenchX.txt.
  - The first and second line should contain the demand of MULT and ALU, respectively.
  - The following i-th line should contain operations scheduled in cycle T_i. List the node indices in ascending order and separate them by blank space.

Homework #1

- Example

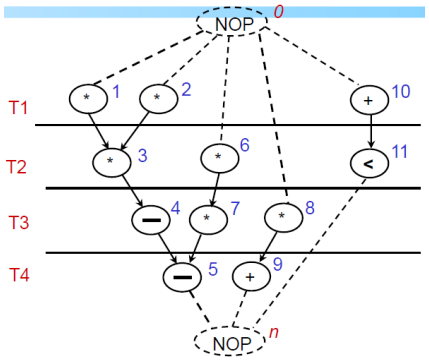
Input File: Bench1.dfg

```
13          → number of nodes is 13
4           → latency constraint is 4
0 0 5 1 2 6 8 10
1 1 1 3
2 1 1 3
3 1 1 4     → Node 3 is a MULT node, it has a successor node, which is node 4
4 2 1 5
5 2 1 12
6 1 1 7
7 1 1 5
8 1 1 9
9 2 1 12
10 2 1 11
11 2 1 12
12 0 0
```



Output File: Bench1.txt

```
2           → demand of MULT is 2
2           → demand of ALU is 2
1 2 10
3 6 11
4 7 8
5 9
```



Homework #1

- Note
  - There are 4 public and 1 hidden benchmarks to evaluate your program.
  - Each benchmark takes 20% of total points.
  - TAs give a score to a benchmark when the result is correct.
  - Please use the following format to run your program:
    - ./LS_StudentID.exe BenchX
    - e.g. ./LS_E12345678.exe BenchX

Homework #1

- Upload data
  - Please upload a .tar file.
  - The .tar file contains a folder which is named by **LS_StudentID.**
  - Put your executable file, source code and header file(If exists) in the folder.
  - If your source code have special requirements, please provide your makefile and readme.