编译原理 – 作业(4)： 代码生成与优化

请大家在复习的时候把作业一到作业四答案都认真对照看一下（我可能有看走眼的地方），另外，平时作业非常重要，请大家认真对待。

截至时间：2023.6.19/周一 23:59:59
提交方式：超算习堂（https://easyhpc.net/course/144）

Q1: 给定如下代码的基本块（Basic Block）：

```
d = b * c
e = a + b
f = a -c
b = b * c
a = e - d
```

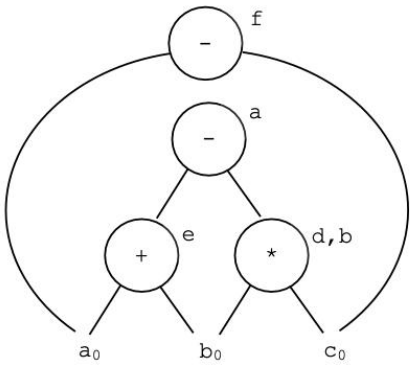
1、构造该基本块的有向无环图(Directed Acyclic Graph，简称 DAG)。
2、分别有如下假设:
   1) 假设#1:仅变量 a 在基本块的出口(exit)是活跃的(live);
   2) 假设#2:变量 f 和 a 在基本块的出口均是活跃的。

试分上述 2 种不同的假设情况，分别基于你构造出来的 DAG 对基本块进行优化。

〖参考答案〗

1、所构造的DAG如下：

2、



假设#1的情况，优化结果为：

```
d = b * c
e = a + b
a = e - d
```

假设#2的情况，优化结果为：

```
d = b * c
e = a + b
f = a - c
a = e - b
```

〖评分标准〗

1、（8分）常见错误之一是未在DAG中合并公共子表达式，扣5分；未标记出初始值结点，扣4分；内部结点有错，每个扣2分。

2、（8分）两种情况分别占4分。优化后的代码未合并公共子表达式，每一假设下均扣2分；计算无活性变量的指令仍存在，每个扣2分；计算有活性变量的指令被删除，每个扣2分；假设#2的优化结果将指令`f = a - c`放在指令`a = e - b`之后，这类指令次序错误每个扣1分。

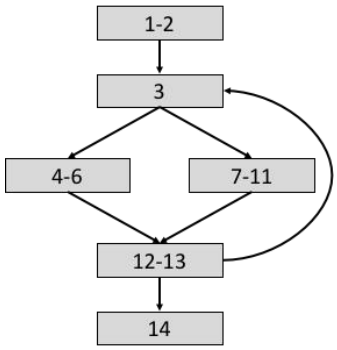
Q2: 给定如下中间代码片段：

```
1:        x = 0
2:        y = 0
3:    L0: if n / 2 goto L1
4:        x = x + n
5:        y = y + 1
6:        goto L2
7:    L1: y = y + n
8:        c = 4 / 2
9:        t1 = x * c
10:       t2 = c - 1
11:       x = x + t2
12: L2: n = n -1
```

```
13:     if n > 0 goto L0
14:     return x
```

1、为上述代码片断划分基本块(basic block)，并画出该代码片断的控制流图(control flow graph，简称 CFG)。你可以直接画出 CFG，在 CFG 的每一结点中用 n-m 表示该基本块由第 n 至 m 条指令组成。

【参考答案】



【评分标准】基本块划分2分，箭头标注2分，每个错误扣0.5分。

2、对第7-11条指令片段，列出两种代码优化方法。

【参考答案】

t1 = x * c是dead code，可以删除

x = x + t2中t2可以通过constant folding和propagation得到，可以替换为x = x + 1

【评分标准】每项1.5分。其中，指出c=4/2→c=2得0.5分，t2=c-1→t2=1得1分。

3、假定所给代码片段（1-14行）来自于函数int Func(int n)，其中n是参数,x、y是局部变量。那么在最终生成的目标代码中，Func被调用时如何访问到n、x和y？提示：函数调用的栈空间由$sp（stack pointer，栈指针）和$fp（frame pointer，帧指针）维护。

【参考答案】

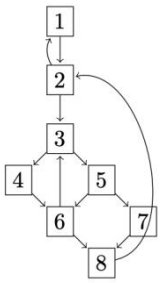
n: $fp + 8

x: $fp – 4

y: $fp - 8

【评分标准】每项1分。n写为$fp+4也视为正确（忽略调用者$fp的保存。

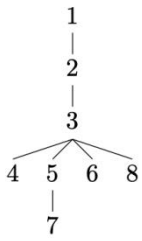
Q4：给定如下控制流图(control flow graph)：[注：需自行学习，详见龙书9.6 Loops in Flow Graphs或链接]



1、节点8的直接支配（immediate dominator）？

Node 3

2、画出该CFG的支配树（dominator tree）；



3、列举出该CFG中的所有自然循环（natural loops），给出循环的头节点和其他节点。

Loop 2 → 1: header node 1, other node 2

Loop 6 → 3: header node 3, other nodes 4, 5, 6

Loop 8 → 2: headernode2,othernodes3, 4, 5, 6, 7, 8