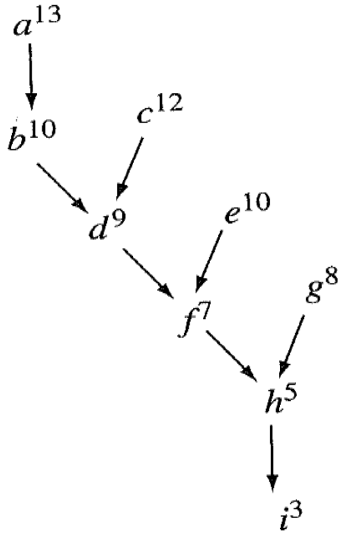
Instruction Scheduling Worksheet

Instructions to be (re)scheduled and corresponding annotated dependence graph:

```
a:  loadAI   rarp,@a ⇒ r1
b:  add      r1,r1   ⇒ r1
c:  loadAI   rarp,@b ⇒ r2
d:  mult     r1,r2   ⇒ r1
e:  loadAI   rarp,@c ⇒ r3 r2
f:  mult     r1,r2   ⇒ r1
g:  loadAI   rarp,@d ⇒ r2
h:  mult     r1,r2   ⇒ r1
i:  storeAI  r1      ⇒ rarp,@a
```



Scheduling algorithm:

```
Cycle ← 1
Ready ← leaves of D  // Max PriorityQueue
Active ← ∅           // Queue

while (Ready ∪ Active ≠ ∅)

   for each op ∈ Active
       if S(op) + delay(op) ≤ Cycle then
          remove op from Active
          for each successor s of op in D
              if s is ready
                 then add s to Ready

   if Ready ≠ ∅ then
       op <- poll(Ready)
       S(op) ← Cycle
       add op to Active

   Cycle ← Cycle + 1
```