5.15

```
void inner4m6(vec_ptr u,vec_ptr v, data_t *dest){
    long i;
    long length=vec_length(u);
    long limit=length-5;
    data_t* udata=get_vec_start(u);
    data_t* vdata=get_vec_start(v);
    data_t sum0=(data_t)0,
        sum1=(data_t)0,
        sum2=(data_t)0,
        sum3=(data_t)0,
        sum4=(data_t)0,
        sum5=(data_t)0;
    for (i = 0; i < limit; i+=6)
    {
        sum0+=udata[i]*vdata[i];
        sum1+=udata[i+1]*vdata[i+1];
        sum2+=udata[i+2]*vdata[i+2];
        sum3+=udata[i+3]*vdata[i+3];
        sum4+=udata[i+4]*vdata[i+4];
        sum5+=udata[i+5]*vdata[i+5];
    }
    for(;i<length;i++){
        sum0+=udata[i]*vdata[i];
    }
    *dest=sum0+sum1+sum2+sum3+sum4+sum5;
}
```

瓶颈可能是由于过度展开导致了寄存器溢出。

5.19

```
void psum1m4a(float a[], float p[], long n)
{
    long i;
    float val, last_val;
    float buf0, buf1, buf2, buf3;
    last_val = p[0] = a[0];

    for (i = 1; i < n - 4; i++)
    {
        buf0 = last_val + a[i];
        buf1 = buf0 + a[i + 1];
        buf2 = buf1 + a[i + 2];
        buf3 = buf2 + a[i + 3];

        p[i] = buf0;
        p[i + 1] = buf1;
        p[i + 2] = buf2;
        p[i + 3] = buf3;
```

```
19
20          last_val = buf3;
21      }
22      for (; i < n; i++)
23      {
24          last_val += a[i];
25          p[i] = last_val;
26      }
```