ICS-QUIZ 2:

Problem 1:

1- True = (a)(b) (d)

2- (c)(e)(f)

3- (c)(d).

4- (a)(b)

5- (b)(d).

6- (a)(b)(c)

7- (b)(d)

8- (a)(b)(c)(d)

9- (e)

10- (a)(b)

Problem 2:

1.
```
int func(int y) {
    int res=0;
    int a= y? (y-1):0;
    int b=0;
    int c=(y+1)? y+1:0;
int pred(int x){
  return (x==0)?0:x-1; //使用 inline 代替 branch
}
int func(int y){
return pred(y)+(pred(0)+pred(y+1)); // reassociation
}
```

2. void sum_rows(double *a, double *b, long n){

long i=0;

for(i=0; i<n; i+=2){ // loop unrolling

double b1=b[i]=0;

double b2=b[i+1]=0; // 减少对同一个地址访问

int m=i*n, p=(i+1)*n; // 将相同结果移出

for(j=0; j<n-1; j+=2) // loop unrolling

{b1 b[i] += a[m+j];

b2 b[i+1] += a[p+j];

}

for(j<n){

b1 b[i] += a[m+j]; b2 b[i+1] += a[p+j];}

b[i]=b1;

b[i+1]=b2;

}

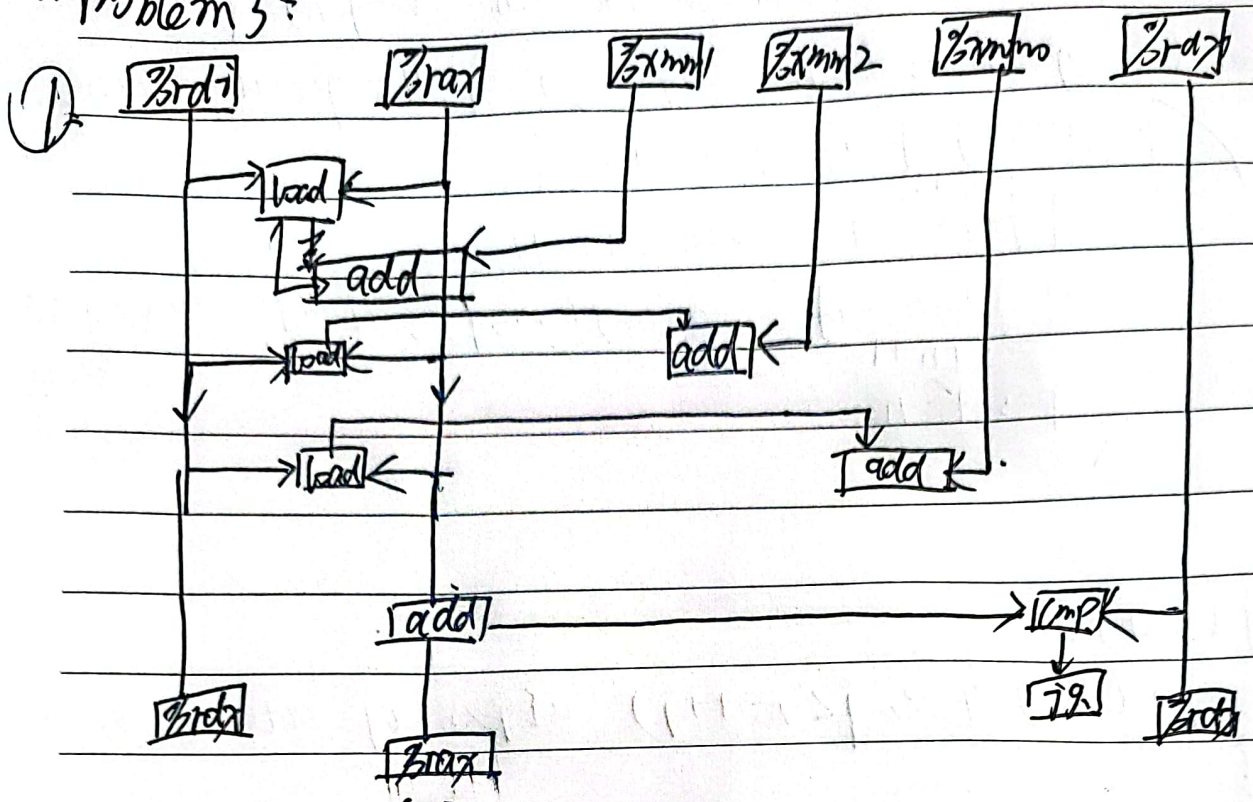
for(i<n){

for(j=0; j<n; ++j) b[i] += a[i*n+j];

}

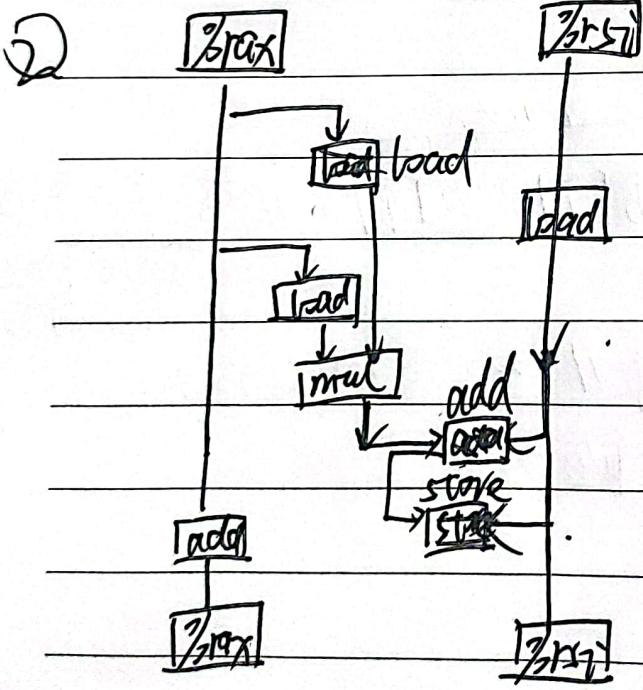
```
void compute(double *a, double *b, long n) {
  for (long i=0; i<n-1; i+=2) { // loop unrolling
    int pos = i*n;
    for (long j=0; j<n; ++j) {
      a[pos+j] = atom2[i,j];
      a[pos+n+j] = atom2[i+1,j]; // 用pos减少乘法次数
      pos += n;
    }
  }
  if (i<n) {
    for (long j=0; j<n; ++j) a[i*n+j] = atom2[i,j];
  }
  for (long i=0; i<n; ++i) b[i] = a[i]; // 将i=0外移,减少比较次数.
  for (long i=1; i<n-2; i+=2) { // loop unrolling
    int pos = i*n; // 减少重复计算
    b[pos] = a[pos];
    b[pos+n] = a[pos+n];
    for (long j=1; j<n; ++j)
      b[pos+j] = a[pos+j] + a[pos-n-1];
  }
  if (i<n) {
    b[i*n] = a[i*n];
    for (long j=0; j<n; ++j)
      b[i*n+j] = a[i*n+j] + a[(i-1)*n+j];
  }
```

Problem 3:

① 

CPE

estimate CPE = 9

② 

critical path:

左边 mul — add — store

因为 store 操作 与 add, mul 的结果存在 data dependency,

estimate CPE = 5 + 1 + 3

= 9

Problem 4:

1. n=1, c='I' \n.
2. n=1, c='S' \n
3. n=1, c='S' \n
4. n=1, c='+' \n
5. output: n=1, c='S' \n
   Tcs.txt中信息：IIS-+IS.
6. ~~n=1, c='S' \n~~ n=1, c='S' \n
7. output: n=0, c='' \n（读到EOF）.
8. n=1, c='S' \n