算法Exercise04

`homework`

本文参考

```
static int n = STATICNUM;
int si[n][2];    //二维数组作为问题求解

//结果状态
int status(int s[2]){
    if s[0] > s[1] return 1
    else return 0
}

//返回最终结果
int* result(int si[][2],int n){
    int wi[n];
    for (int i=0;i < n;i++){
        wi[i] = status(si[i]);
    }
    return wi;
}

//l,r分别为骨片序号
void sort_domino(int si[][2],int l,int r){
    L[1..n] <- si[][0]; //左边放在L数组中
    R[1..n] <- si[][1]; //右边放在R数组中
    if (r - l) == 0 then return; //骨牌数不能少于两个
    L[r+1] = 0;
    R[l-1] = 0;
    int max = max_domino(L,R,l,r);
    si[][0] <- L[1..n];
    si[][1] <- R[1..n];
}

int max = 0;
int max_domino(int L[], int R[], int l, int r){
    if (r - l) == 1 return R[l]*L[r];
    else
        mid = (l+r)/2;
```

```
            max1 = max_domino(L,R,l,mid) + max_domino(L,R,mid,r);
            swap(L[mid],R[mid]);     //交换被拆分的骨牌左右半块
            max0 = max_domino(L,R,l,mid) + max_domino(L,R,mid,r);
            if max0 < max1 {
                swap(L[mid],R[mid]);
                max = max1
            }else{
                max = max0;
            }
            return max;
    }

    void main(){
        int l = 0;
        int r = n-1;
        sort_domino(si,l,r);
        int *p = result(si,n);
    }
```