Homework 4

Algorithm Design

- 6-1

  (a) 5—3—3—4—3 通过算法得到最大独立集为 9，但实际的最大独立集为11

  (b) 10—1—1—9 通过算法得到最大的独立集为 11，但实际的最大独立集为19

  (c)

  ```
  int findMax(int num[]){
      int preMax = 0;
      int currMax = 0;
      for(int x : num){
          int temp = currMax;
          currMax = Max( prevMax + x, currMax );
          prevMax = temp;
      }
      return currMax;
  }
  ```

  $f(k) = max(f(k-2)) + A_k, f(k-1)$

  复杂度分析：

  时间复杂度：$O(n)$

  空间复杂度：$O(1)$

- 6-2

  (a)

	Week 1	Week 2	Week 3
l	2	2	2
h	1	5	10

  通过算法得到的最大利益为7，但实际的最大利益为12

  (b)

  ```
  int findMax(int L[], int H[]){
      L(1) = L[1];
      H(1) = 0;
      loop:
          L(i) = L[i] + max(L(i-1), H(i-1));
          H(i) = H[i] + max(L(i-2), H(i-2));

      return max(L(i), H(i));
  }
  ```

- 6-11

```
int findMin(int s[]){
    OPT(1) = s[1];
    while(i < num){
        OPT(i) = min(rs[i]+OPT(i-1), 4c+OPT(i-4));
        i++;
    }
    return OPT(num);
}
```

- 6-13

  we build a graph with the nodes means each stock, and the directed edge(i, j) for each pair of stocks. we put a cost of $-log_{r_{ij}}$ on edge (i, j).

  A trading cycle C in G is an opportunity cycle iff $\prod_{(i,j)\in C} r_{ij} > 1$ or iff $\sum_{(i,j)\in C} log_{ij} > 0$

  A trading cycle C in G is an opportunity cycle iff it is a negative cycle. Thus we can use the polynomial-time algorithm for negative-sycle detection to determine whether an opportunity cycle exists.