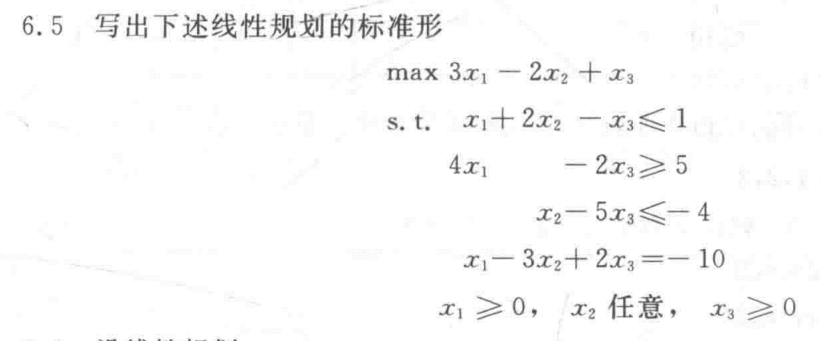
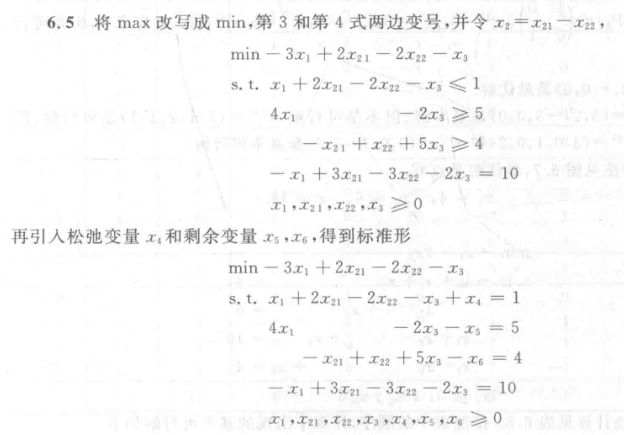
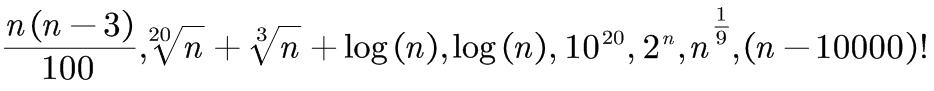
第一题题目：



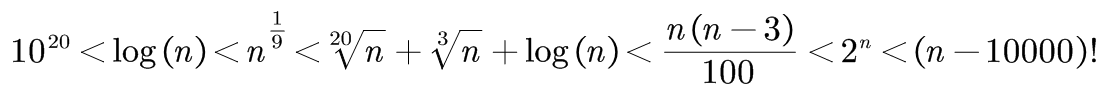
答案：



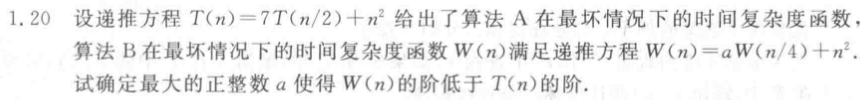
第二题题目：



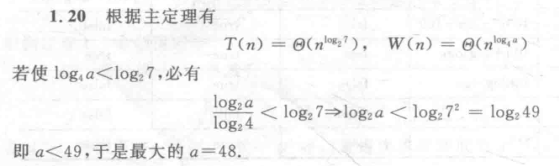
第二题答案：



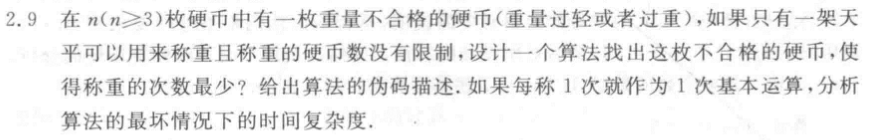
第三题题目：



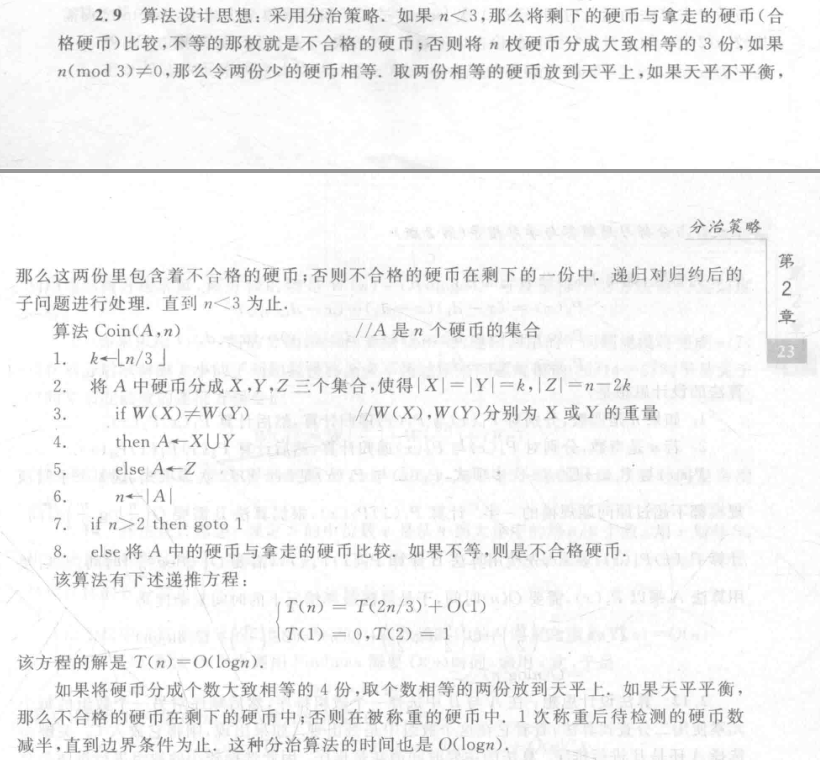
第三题答案：



第四题题目：



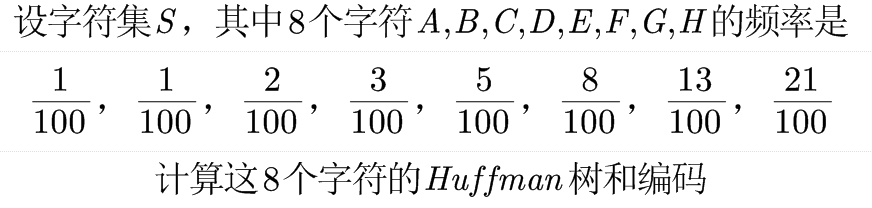
第四题答案：



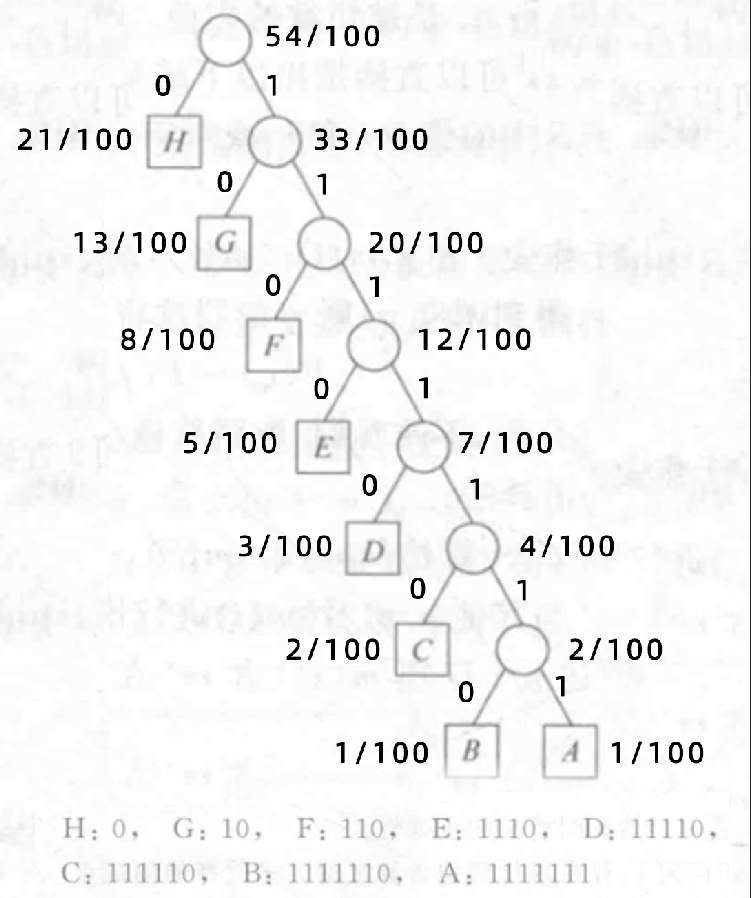
5.use dynamic programming to find the lengthest sub-sequence in order of increasing in A=<1,19,-4,8,3,11,5,-7,9,6,13>

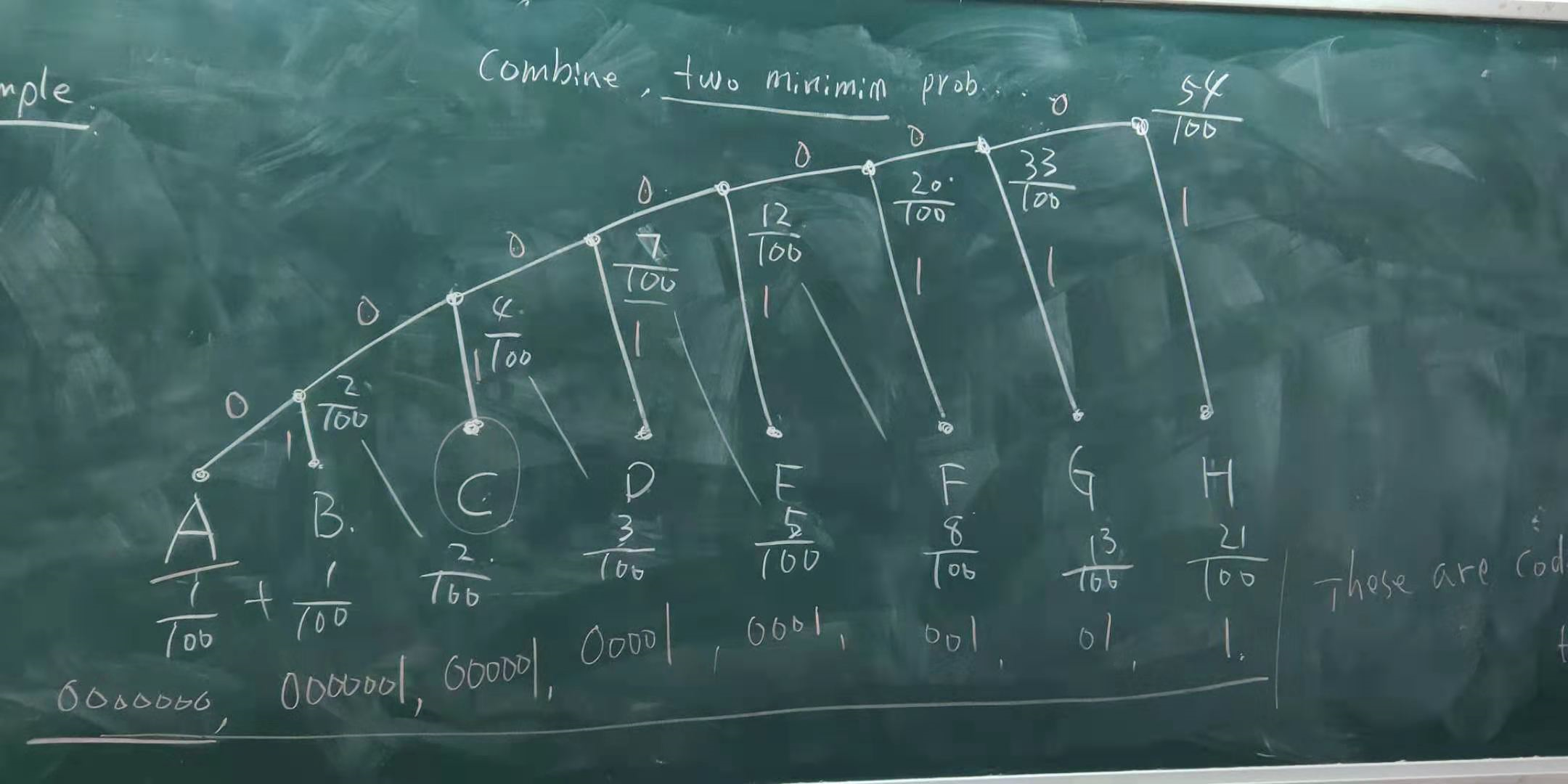
（后面接了个so simple）

第六题题目：



第六题答案：（下图和黑板上写的编码方式不太一样，同时附上黑板的）





7.summarize the main idea,computing stages,and notices, of Backtracking AlogorithmP119

8.summarize the complexity of Sorting Algorithm,good algorithm,and problem complexity.Further illustrate the relationship among the above three aspects with over 200 words

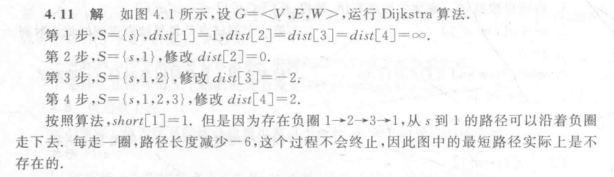
9.概述回溯算法的有关概念（怀疑和7.是一样的题）

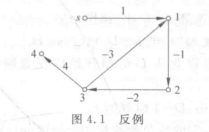
* 回溯算法（what）
* 解空间（what）
* 代价函数（what+why）
* 界函数（what+why）
* 多米诺性质（what）

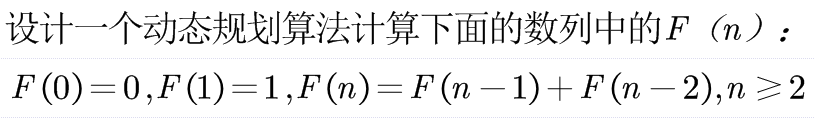
第十题题目：



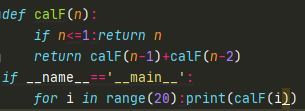
第十题答案：





11.

师兄写了个递归算法，但我认为不应该那样做的（所以我就先把师兄的贴上来有空再回来更新）



12.设计一个算法，对于一个给定的包含n个整数的集合S和另一个给定的整数x，该算法可以在O(nlogn)时间内确定S中是否存在两个元素，使得它们的和恰为x

（网上随便copy的，还没确认，仅供参考）

#include <cstdio>
  
#include <cstdlib>
  
#include <cstring>
  
//快排 时间复杂度O(nlogn)
  
void quick\_sort(int\* arr, int l, int r) {
  
 if (l >= r) return;
  
 while (l < r) {
  
 int x = l, y = r;
  
 int z = arr[l];
  
 do {
  
 while (x <= y && arr[x] < z) x++;
  
 while (x <= y && arr[y] > z) y--;
  
 if (x <= y) {
  
 int temp = arr[x];
  
 arr[x] = arr[y];
  
 arr[y] = temp;
  
 x++, y--;
  
 }
  
 } while (x <= y);
  
 quick\_sort(arr, l, y);
  
 l = x;
  
 }
  
}
  
//判断是否满足条件 O(n)时间复杂度
  
bool is\_ok(int\* A, int x, int len) {
  
 quick\_sort(A, 0, len - 1);
  
 int i = 0, j = len - 1;
  
 while (i < j) {
  
 if (A[i] + A[j] == x) return true;
  
 else if (A[i] + A[j] > x) j--;
  
 else i++;
  
 }
  
 return false;
  
}
  
   
int main() {
  
 int A[8] = {3, 41, 52, 26, 38, 57, 9, 49};
  
 int x;
  
 while (scanf("%d", &x) != EOF) {
  
 printf("%d\n", is\_ok(A, x, 8));
  
 }
  
   
 return 0;
  
}