

判断题：

1. ×
2. √
3. ×
4. √
5. ×
6. √
7. √
8. ×
9. ×
10. √

选择题：

1. B
2. A
3. D
4. A
5. C
6. A
7. B
8. C
9. B
10. D

简述：（见书）

1. 在执行过程中可随机性地选择下一个计算步骤的算法称概率算法。分为数值概率算法、蒙特卡洛算法、拉斯维加斯算法和舍伍德算法。
2. 将序列5个5个分组，取出每组的中位数，再取出所有组中位数的中位数，以此作为pivot进行一次划分，保证每次划分的子数组长度至少为原数组的 $\frac{3}{4}$ 大小。
3. 算法指解决问题的方法或过程。满足确定性、有限性和可行性。

解答：

一、

(1): 分为999, 再分为333, 再分为111。（简述了, 懂就行）

(2): 分为 $3^{k-1} \times 3$, 重复(1)的过程。算法略。

复杂度: $T(3^k) = T(3^{k-1}) + O(1) = kT(1) + O(1) = O(\log_3 n)$

二、

(1): 设物品编号1-5, 从1开始往5选。

①: (0,0)(5,3)

②: (0,0)(5,3)(12,4)(17,7)

③: (0,0)(5,3)(6,7)(11,10)(18,11)

④: (0,0)(5,3)(6,7)(7,9)(11,10)(12,12)(13,16)(18,19)

⑤: (0,0)(5,3)(6,7)(7,9)(11,10)(12,12)(13,16)(18,19)

(2): 最优解{1,3,4}, 最优值为19

三、

(1) $\{x_1, x_2, \dots, x_n\}$, x_i 表示在第 i 行的皇后放在第 x_i 列处。

(2) 显约束: $x_i \in [1, n]$

隐约束: $x_i \neq x_k$ 和 $|x_k - x_i| \neq |k - i| (i \neq k)$

(3): 我没看懂这个题, 我的理解是一个满四叉树 (深度为4)

四、

(1): 一直-3, 剩4个就变成2+2。

(2): 易证, 略。

五、

这个题我其实挺懵的, 可以好好交流下, 我动态规划并不好。

(1):

$c[i, j] = \min(c[i, j - x[k]] + 1, c[i - 1, j]) (k \in [1, i], \text{且} \forall k, j - x[k] \geq 0)$

初始化: $c[0, k] = k, c[m, 0] = 0, k \in [0, j], m \in [0, i]$

(2): 假设矩阵为 $(j+1) \times (n+1)$, 在这题中为 5×9 , 矩阵见下:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \\ 0 & 1 & 1 & 2 & 1 & 2 & 2 & 3 & 2 \\ 0 & 1 & 1 & 2 & 1 & 1 & 2 & 2 & 2 \end{pmatrix}$$

(3): 见代码, 下为网址和源码

paste.ubuntu: <https://paste.ubuntu.com/p/DgTTjXW4qV/>

```

1  #include <iostream>
2  using namespace std;
3  int c[1010][1010];
4  int n, m;
5  int x[1010];
6  void solve(){
7      for(int i = 0; i <= n; i++){
8          c[i][0] = 0;
9      }
10     for(int i = 0; i <= m; i++){
11         c[0][i] = i;
12     }
13     for(int i = 1; i <= n; i++){
14         for(int j = 1; j <= m; j++){
15             int mink = 0x3f3f3f3f;
16             for(int k = 1; k <= i; k++){
17                 if(j - x[k] >= 0 && mink > c[i][j-x[k]])
18                     mink = c[i][j-x[k]];
19             }
20             c[i][j] = min(mink+1, c[i-1][j]);
21         }
22     }

```

```
23 }  
24  
25 int main(){  
26     cin >> n >> m;  
27     for(int i = 1; i <= n; i++) cin >> x[i];  
28     solve();  
29     cout << c[n][m];  
30     return 0;  
31 }
```