Q1

$$对一个随机输入的数组，0<\alpha\le 1/2$$
$$若PARTITION产生比1-\alpha:\alpha更差的划分$$
$$需要选定的主元（pivot\ element）在最小的\alpha n个元素或者在最大的\alpha n个元素当中$$
$$这时对主元的选取概率近似为\frac{\alpha n}{n}\times 2=2\alpha$$
$$也就是说，如果要产生更好的划分的概率为1-2\alpha$$

Q2

(1)

$$当x为\{a_n\}中位数时，minD(x)最小$$
$$而D(x)=\frac{\sum_{i=1}^{n}w_i|a_i-x|}{\sum_{i=1}^{n}w_i}对\forall给定x,求解的时间复杂度为\Theta(n)$$
$$因而，问题转化为，求一个O(n)的寻找中位数的算法$$
$$1.将输入数组a_n的n个元素划分为\lfloor n/5\rfloor组，每组5个元素，且至多只有一组由剩下的nmod5个元素组成$$
$$2.寻找这\lceil n/5\rceil组中每一组的中位数：首先对每组元素进行插入排序，然后确定每组有序元素的中位数$$
$$3.对第2步中找出的\lceil n/5\rceil个中位数，递归调用算法寻找中位数x$$
$$4.按中位数的中位数x对a_n进行划分，让k比划分的低区中的元素数目多1，因此x是第k小的元素。并且由n-k个元素在划分的高区$$
$$5.如果k=\lfloor n/2\rfloor，返回x$$
$$如果k>\lfloor n/2\rfloor,在低区递归调用找第\lfloor n/2\rfloor小元素$$
$$如果\lfloor n/2\rfloor>k，在高区递归调用查找第i-k小的元素$$

(2)

$$由题意：\Delta\{a_n\}=\Delta^2\{b_n\}$$
$$有：\{a_{n+1}-a_n\}=\{b_{n+1}-2b_n+b_{n-1}\}$$
$$根据差分的性质可得：b_i=\sum_{k=1}^{i}a_k$$
$$a_i=b_i-b_{i-1}$$
$$用两个数组A[T]和B[T]存储a_n的一阶差分和b_n的二阶差分的值$$
$$用两个指针i和j指向数组的开头$$
$$初始化一个变量min,初始化为+\infty$$
$$开始进入循环diff=A[i]-B[j],如果diff比min小，更新min为diff$$
$$diff<0,i++$$
$$diff>0j++$$
$$diff=0,返回min为所求$$

Q3

(1)

$$A和B事严格升序的，不存在i，h\ s.t.\ A[i]=B[j]$$
$$S=\{<i,j>|A[i]>B[j]\}$$
$$1.采用两个指针i和j分别指向A和B的起始位置A[0]和B[0]$$
$$用计数器cnt统计|S|,并初始化为0$$
$$2.如果A[i]>A[j],cnt++,若j<m,j++$$
$$如果A[i]<A[j],若i<n,i++$$
$$3.当i=n且j=m时，算法结束，输出cnt即为|S|$$

(2)

$$给定长度为n的数组L，数组内无重复元素，定义I=\{<i,j>|i<j\bigwedge L[i]>L[j]\}$$

$$即求一个数列中逆序对的数量，思路为修改过的归并排序$$

$$1.规定res为逆序对的数量，初始化为0，我们要统计三部分逆序对$$

$$l->mid内的所有逆序对$$

$$mid+1->r内的所有逆序对$$

$$两个区间交叉部分的逆序对$$

$$2.对l->mid内的逆序对，递归调用res=merge_sort(q,l,mid)即可$$

$$3.对mid+1->r的逆序对，递归调用res=merge_sort(q,mid+1,r)即可$$

$$4.对于交叉部分的逆序对,在合并数组时$$

$$while(i<=mid\&\&j<=r)$$

$$\{$$

$$if(q[i]<=q[j])\ temp[k++]=q[i++];$$

$$else\{$$

$$temp[k++]=q[j++];$$

$$res+=mid-i+1;$$

$$\}$$

$$\}$$

$$因为这时候在找到一个q[i]>q[j]的情况下，有序的前半数组剩下的所有元素都比q[j]大$$

$$5.最终res的结果即为逆序对的数量，而且归并排序是O(nlgn)的$$

(3)

$$现在有两个不含重复元素的数列A_1和A_2$$

(4)

Q4

(1)

算法1：

完备的，证明如下

算法从头开始，对1有$1 \rightarrow n$，有n个位置可能被选中

算法存在一种情况，2不选择1已经选择的位置，因此可以有$n-1$个位置选择，同理，对3，可以有$n-2$种位置选择$...$

满足$n!$中排列的方式，因此是存在全排列的可能，因而算法是完备的

算法2：

对1是否与j位置交换的概率分别为$\frac{1}{2}, \frac{1}{2}^2 \ldots \frac{1}{2}^n$，也即$n$个位置都可以取到

对2是否与j位置交换的概率同理，同时也可以做到不与1的当前位置交换，因此可以取到$n-1$个位置

$...$

即依然有$n!$种排序的可能

故该算法是完备的

算法3：

算法是完备的

假设第一次选定元素x_1，x_1有与n个元素交换的可能，也就是有n个位置

在循环迭代过程中，第二次可以选中$x_2 \neq x_1$，且x_1有$n-1$个位置

$...$

最后，x_n可以取到不同于x_i，$i=1,...,n-1,$且只有一种可能，因此是$n!$种不同的排列

因而我们可以认为，该算法是完备的

算法4：

该算法是完备的

第一次迭代中，索引i为1，j的范围是从1到n，即可以选择数组中任意一个索引与位置1的元素进行交换

第二次迭代中，索引i为2，j的范围是从2到n，即可以选择数组中除了位置1的生元素与位置2的元素进行交换

以此类推，每次迭代中，索引i的范围是从i到n，可以选择数组中任意一个索引与位置i的元素进行交换

由于每次迭代都可以选择任意一个位置的元素与当前位置i的元素进行交换，因此在n次迭代后，可以生成给定数组的所有可能排列

(2)

算法1：

不均匀

对$i=1$来说，n个位置是等可能的，但对后续$2,3...n$来说，位置1是不可选的，因此不是等概率的

算法2：

均匀

$randint(0,1)$等概率生成0和1的随机数，因此每次交换操作被执行的概率都是相等的

每个迭代中，外部循环i遍历了1到n的所有可能取指，而内部循环变量j也遍历了1到n的所有可能取指

因此，对于数组任意一对元素，交换操作被执行的概率是相等的。

因此，算法是均匀的

算法3：

不均匀

两个随机索引的生成是独立的，同一对索引可能被多次生成，并且同一个索引可能在同义词迭代中被生成两次

某些元素会被交换多次，有些元素不会被交换

算法4：

均匀

第一次迭代中，索引i为1，j的范围是1到n，即可以选择数组中任意一个索引与位置1的元素交换

第二次迭代中，索引i为2，j的范围是从2到n，可以选择数组中除了位置1的剩余元素与位置2的元素进行交换

以此类推，每次迭代中，索引i的范围是从i到n，可以选择数组中任意一个索引与位置i的元素交换

由于每次迭代都可以选择任意一个位置的元素与当前位置i的元素进行交换，n次迭代后，可以生成所有等可能排列，因而是均匀的