第一题

树形DP，设$f(u)(x)$为u取值x时其子树形成小根堆的概率密度。有转移

$$f(u)(x) = \frac{[x \leq b_u]}{b_u} \prod_{v \in \mathrm{son}(u)} \int_x^{+\infty} f(v)(t)\mathrm{d}t$$

归纳可知$f(u)$是一个限制在某个$[0, c]$上的多项式，因此积分乘法都很好做。暴力DP复杂度大概就是$O(n^3)$。

第二题

先把所有比赛按照$\frac{a_i}{b_i}$从大到小排序。

因为b小a大，所以状态中只能记录有关b的那一部分。直接DP很难做到这样的效果，如果要只和b有关就必须支付更多的复杂度计算一些冗余状态以达到不重不漏的目的。

不妨直接大胆枚举后一半的b之和，然后在计算出的结果中只挑选b之和等于所枚举的答案的那一部分。

$dp_{i,j,k}$表示确定了前i个比赛是放在前面还是后面，其中j个比赛是放在前面的，剩下的$i-j$个放在后面的比赛的b的总和是k，此时的最大收益。

枚举下一个是放在前面还是后面来进行转移，具体细节可以看代码。

复杂度$O(n^4b^2)$。期望得分100。

第三题

考虑用Kruskal求MST。也即，要按照两两之间的编辑距离来排序。

把$[0, 2^n]$之间的所有串和与它编辑距离是1的串连边。那么x, y之间的编辑距离就是这张图上的最短路。不妨把所有m个点作为起始点入队，然后在这张图上bfs。

记录$f(u)$表示第一个bfs到u的起始点。这样每当一个点v更新到u，那么就把$f(v)$和$f(u)$作为一个候选来更新一下。但是注意到路径长度为奇数和偶数的区别，想要线性做这一部分需要一点小小的特判。